

AD-A137 110

EXTENDING GRAMMARS TO NEW DOMAINS(U) UNIVERSITY OF
SOUTHERN CALIFORNIA MARINA DEL REY INFORMATION SCIENCES
INST J J ROBINSON JAN 84 ISI/RR-83-123
F49620-79-C-0181 F/G 6/4

1/1

UNCLASSIFIED

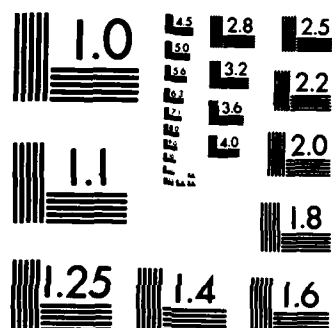
F/G 6/4

NL

END

CHINESE

OTC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A137110

Jane J. Robinson
Consultant

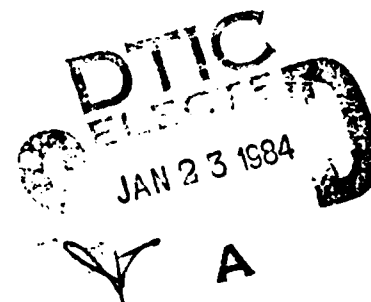
ISL/RR-83-123
January 1984

University
of Southern
California



Extending Grammars to New Domains

DTIC FILE COPY



INFORMATION
SCIENCES
INSTITUTE



213/822-1511

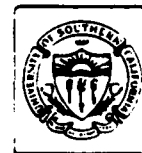
4676 Admiralty Way / Marina del Rey / California 90292

84 01 20 029

ISI/RR-83-123

January 1984

University
of Southern
California



Jane J. Robinson
Consultant

Extending Grammars to New Domains

INFORMATION
SCIENCES
INSTITUTE



A-1

213/822-1511

4676 Admiralty Way / Marina del Rey / California 90292

This material is based on work supported by the National Science Foundation under Grant No. MCS78-07410. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of the National Science Foundation.

Supplementary funding was provided by the Air Force Office of Scientific Research Contract No. F49620-79-C-0181

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 68 IS OBSOLETE
S/N 0102-014-6601

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (continued)

This is the report of an undertaking to extend and adapt an existing grammar, DIAGRAM, to provide the syntactic analysis of sentences in a new domain. DIAGRAM is an augmented phrase-structure grammar whose rules provide a means for associating semantic and domain dependent interpretations with a syntactic analysis. An earlier version, used for the syntactic analysis and the interpretation of spoken English, covered the vocabulary and basic phrase types needed to query a static data base of information about naval ships.

The new domain in which the extended and adapted version has been tested is represented by a set of eighteen dialogues, called Helper dialogues, in which computer users present their problems to the operator and ask for help. Extending the syntax to cover the new words and phrase types exhibited in these sample texts raises a number of questions of general theoretical interest along with problems that can properly be construed as artifacts of the particular grammar that is being extended or of the limitations of the parsing program and the computer system in which the grammar is applied to input sentences. This report therefore can be read with both a broad and a narrow scope. The narrow scope reading is concerned with the additions and revisions that were made to DIAGRAM in order to parse the sentences in the dialogues selected from the new domain. The broad scope reading is concerned with the kinds of problems encountered in extending syntactic coverage generally and with strategies for coping with them.

Contents

1. Introduction	1
2. The process of extending DIAGRAM	2
2.1 Overview	2
2.2 Summary of additions and modifications	3
2.3 Summary of test results	5
2.4 Analysis of failures	6
3. Procedures, problems and policies	8
3.1 Purging a grammar	9
3.2 Adding vocabulary	11
3.3 Folding the new vocabulary into the rules	13
4. Summing up and pointing ahead	15
I. The four dialogues used for developing the grammar	16
II. The fourteen additional dialogues used for testing the grammar	18
III. The final rules	24
References	58

1. Introduction

This is the report of an undertaking to extend and adapt an existing grammar, DIAGRAM, to provide the syntactic analysis of sentences in a new domain.¹ DIAGRAM is an augmented phrase-structure grammar whose rules provide a means for associating semantic and domain dependent interpretations with a syntactic analysis [Robinson 82]. An earlier version, used for the syntactic analysis and the interpretation of spoken English, covered the vocabulary and basic phrase types needed to query a static data base of information about naval ships [Walker 78]. The sentences appearing below illustrate the typical queries that DIAGRAM and the system in which it was embedded were prepared to both parse and interpret.

What is the draft of the J.F. Kennedy?
Is its speed greater than thirty knots?
Who is the commander?
How many ships are within two hours steaming time of the Lafayette?
Which ships have doctors on board?

DIAGRAM was subsequently redesigned for typed input and its syntactic coverage was broadened. However, semantic interpretations were not associated with all of its rules, several minor syntactic categories were represented only by placeholders, and many complex syntactic constructions were omitted. It was tested primarily on those rules that did include semantic interpretation procedures for answering questions. Consequently, the syntactic coverage of the grammar as a whole was never systematically tested in other domains and in communicative tasks conducted in give-and-take conversational mode.

The new domain in which the extended and adapted version has been tested is represented by a set of eighteen dialogues, called Helper dialogues, in which computer users present their problems to the operator and ask for help. The group of sentences appearing below were selected from these dialogues. Comparison with the previous group of sentences will convey a sense of the differences in vocabulary and syntax required for each domain.

On Wednesday, I created a rather lengthy listing on the line printer.
I'll pick it up this afternoon.
You don't have access to Devtenex, do you?
Can I recover File.name from the most recent system dump?
What tape number do you want me to mount?
Did you delete a file accidentally, and you want it back?
I'll see if I can restore it; however, I don't know how soon.
Pick the highest version, if there are more than one.
How about Tape 247?

¹This report represents grammar development work done principally in 1980-81. The report was substantially complete in 1982, but its release has been delayed beyond the author's control. -- W.M.

In addition to differing in vocabulary, as one would expect from the differences in domains, the second group is much more colloquial than the first and is also much more complex syntactically. Extending the syntax to cover the new words and phrase types exhibited in these sample texts raises a number of questions of general theoretical interest along with problems that can properly be construed as artifacts of the particular grammar that is being extended or of the limitations of the parsing program and the computer system in which the grammar is applied to input sentences. This report therefore can be read with both a broad and a narrow scope. The narrow scope reading is concerned with the additions and revisions that were made to DIAGRAM in order to parse the sentences in the dialogues selected from the new domain. The broad scope reading is concerned with the kinds of problems encountered in extending syntactic coverage generally and with strategies for coping with them.

2. The process of extending DIAGRAM

2.1 Overview

The description that follows assumes some familiarity with the augmentation of context-free rules employed in DIAGRAM, particularly with the use of constructors to propagate attributes of constituents and to use attributes to reject or evaluate syntactic combinations.²

The general procedure in extending DIAGRAM was to revise and augment its lexical entries and rules to cover four dialogues and then test it on an additional fourteen dialogues, adding lexical entries for new words but without touching the rules.³ (An attempt was made initially to have non-linguists add the vocabulary of the four tuning dialogues, but this proved impractical.) Note that this extension did not involve providing semantic and pragmatic interpretations. This meant that no semantic tests could be used to control the number of syntactic analyses the grammar produced. However, the success of the extension was judged partly on the grounds of whether or not the rules were able to indicate which analyses were syntactically the most likely when many parses were produced. For example, in "give the book to her", the analysis that attaches "to her" as part of the verb phrase "give....to her" is more likely than the one that attaches it as part of the noun phrase "the book to her", although the latter is syntactically possible - (cf. "the key to the apartment".) The formalism for writing the rules permits making such judgments through the assignment of numerical scores to analyses. Semantic interpretation routines may be applied only to the analyses with the

²We gratefully acknowledge the active participation of James A. Moore and William C. Mann in this work.

³The dialogues appear in the appendices. All testing was done with the "translators" and "integrators" (described in other documents on DIAGRAM) turned off.

highest scores. The intent in providing this kind of constraint is to allow the grammar to recognize legitimate sentences without having to interpret semantically every possible but improbable reading. Without this capability, it can easily become impossible to test a grammar, because the parser will overwhelm the system with multiple analyses. This is an area in which little research has been done and few concrete proposals to apply likelihood constraints have been tested on any significant scale since the phase-out of the ARPA-sponsored research on understanding continuous speech.

2.2 Summary of additions and modifications

During the first phase of adapting DIAGRAM, before "freezing" it and testing it on the additional fourteen dialogues, no count was kept of the words added. However, it was during this phase that the crucial and difficult additions to the lexicon were made. Lexical additions included a number of verbs that required extensive modifications to the rules for verb phrases, but by far the most difficult additions were words that did not fall easily into any of the major categories of noun, verb, and adjective. New categories were created and adjustments were made to old categories and rules. All in all, four new categories and twentyone new rules were introduced. The changes are shown in Table 2-1.

Three of the new categories, TAGWORD, TAG, and RESPONSE, were added to handle elliptical conversational phrases, including the ubiquitous "OK". Other examples include "How about Tape Tape2?" and "Probably." Adding these categories entailed adding four new rules. These additions are not sufficiently general, and any sustained attempt to handle ellipsis will require special mechanisms. I would advise reworking them if any further revision is contemplated, or purging them (see Section 3.1) if DIAGRAM is to be extended to yet another domain with a different style of discourse.

Two rules were also added to handle conjunction of adjective and verb phrases. (The grammar already contained some provision for conjoining nominals and sentences.) Here too the rules are not sufficiently general. There is one compelling reason for not trying to handle all possible combinations allowed in conjunctions solely by syntactic rule. Unless semantic and pragmatic constraints can be imposed, or unless special mechanisms can be developed to suspend the regular parsing procedure when a conjunction is encountered, the combinatorial explosion among the number of parsings is catastrophic. Consequently, it was decided to add only enough conjunction capability to handle the sentences in the tuning dialogues. Otherwise, no testing would be possible within any reasonable resource limits of time and storage capacity.

Table 2-1: Differences Between Original and Final Grammars

	OLD ---	NEW ---
CATEGORIES	81	78
RULES	101	108
CATEGORY CONSTRUCTORS	2	5
RULE CONSTRUCTORS	86	92

	NUMBER -----
ADDED CATEGORIES	4
DELETED CATEGORIES	7

ADDED RULES	21
CHANGED RULES	46
DELETED RULES	14

ADDED RULE CONSTRUCTORS	17
ADDED CATEGORY CONSTRUCTORS	3
CHANGED RULE CONSTRUCTORS	58
DELETED RULE CONSTRUCTORS	11

A new category, GERUND, was added by rule to handle verbal nouns, that is, noun phrases created by added "-ing" to verbs in verb phrases. Gerunds are responsible for the kind of ambiguity exhibited in the classic "Flying planes can be dangerous," which can mean either that the planes themselves are dangerous when they are flying or that flying them is dangerous. This addition entailed modifications to many other rules in order to control possible explosions in the number of analyses.

Thirteen rules were added to handle adverbial expressions, including subordinate sentences. Most of the changes in the original rules were made to accommodate these additions. This is the area in which the original grammar was most drastically changed. Adverbial expressions include prepositional phrases, and these, along with conjunctions, are the most notorious sources of combinatorial explosions. It is not surprising, therefore, that most of the original rules had to be modified when adverbial expressions were added. The effects when conjunctions and adverbial expressions combine in a single sentence can be startling. A short sentence like "Let me look for it and get back to you, ok?" strained the system's storage capacity to the limit, even though many constraints were introduced into the rule constructors where modifications took place.

The added constraints were of several kinds. Some were tests for syntactic well-formedness. Occasionally, as in the case of "time" adverbials, the tests bordered on semantics, and were essentially checks for selectional restrictions. The subtlest of the constraints, and the most difficult to specify accurately, were those that did not straightforwardly accept or reject the constituent analysis proposed by a rule, but gave it a score that indicated its likelihood. As pointed out in the overview (2.1), if DIAGRAM provides for more than one analysis of a sentence, it is possible to discriminate among competing analyses by their scores. The addition of gerunds motivated one such constraint. For example, "Swimming is my favorite sport" is analyzed as a sentence whose subject nominal is a gerund. However, while sentences like "My favorite sport is swimming" are possible, in which the gerund is in predicate nominal position, its occurrence in that position is considerably restricted, since the subject NP must denote an activity. It is more likely therefore that "swimming" (and other V + ing occurrences) will be verbs, as in "John is swimming."

2.3 Summary of test results

The grammar was tuned until it could successfully parse all sentences in the four original dialogues. It not only provided the correct parses for each sentence, but with one exception, it also scored them correctly, according to their likelihood. It was then applied to the additional fourteen sentences with no further tuning. Although lack of rigorous criteria for success means that numerical analyses of results can be only approximate, it seems clear that the tuned grammar was remarkably successful in producing correct analyses for most of the sentences in the new dialogues, evaluating them correctly with respect to their various likelihoods and constraining the analyses so that the system was not overwhelmed by incorrect analyses.

In preparing for the final test, approximately three hundred words had to be added to cover the fourteen new dialogues. Nouns, verbs, and adjectives accounted for two-thirds of the additions. The breakdown is, approximately 100 new nouns, 75 new verbs, and 25 new adjectives. These ratios are

to be expected. The fact that most of the additions after an initial tuning will be from these content-word classes accounts in part for the possibility of successful extension of a grammar to new domains. These classes are the least likely to require highly specialized rules and constraints.

Test results are summarized by Moore, who partitions the responses of the grammar to each sentence in the dialogues into 13 categories. They include cases in which the grammar could not be expected to parse the sentence because the sentence itself was clearly ungrammatical. Although some effort had been made to normalize the sentences before testing, several sentences escaped editing, and one entire dialogue (OC549) was tested in an unedited version. An edited version of another dialogue (OC193) contained the sentence "Well, where it's located, would a terminal be available for a traveler, who to contact in London once there, etc.", which, not surprisingly, failed to parse. Ignoring these cases and eliminating the trivial sentences such as "OK", he compares the results for the original grammar applied to the tuning dialogues with the final "frozen" grammar as applied to the tuning dialogues and to the test dialogues. The results are:

	Wins	Loses
First 4 dialogues, Original grammar:	36.4%	63.6%
First 4 dialogues, Augmented grammar:	95.7%	4.3%
Final 14 dialogues, Augmented grammar:	66.0%	34.0%

(From Jim Moore)

Moore goes on to say: "The final observation is that the augmentations (to accommodate the first four dialogues) did not cripple the grammar for subsequent use. On the contrary, the augmented grammar produced twice the percentage of wins and half the percentage of loses, compared with what would be expected from the exhibited performance of the original grammar."

2.4 Analysis of failures

In order to prevent overlong attempts to parse, a storage resource limit was set at 600 for the number of phrases that could be constructed during parsing of a single sentence. Sentences that did not parse within that resource limit were counted as failures. No complete, detailed analysis of the causes of all their failures was made. However, examination indicates that some would have parsed successfully, had the limit been raised only slightly. Such sentences tended to be very long or to contain conjunctions and several prepositions. Some combined all three properties.

Inspection shows that many failures were due to oversights that I judge to be relatively easy to

correct. That is to say, correcting them would probably not have ripple effects in interacting with the rest of the grammar.

Some failures involved lexical entries where words were not coded correctly and therefore lacked properties that allowed appropriate rules to apply. For example, "go" was not coded to allow it to accept an infinitive object ("going to go"); "until" was not coded as a subordinating conjunction ("until I can find it"); "another" was not coded as an indefinite determiner ("another copy").

Some rules were too restrictive. Thus, indefinite pronouns like "someone" were not permitted to have postmodifiers ("someone who knows about it"). This restriction alone accounted for three failures and seems easily correctable.

Some rules contained errors in their constructors, so that they rejected legitimate parsings. For example, one of the rules for relative clauses rejected a legitimate parse on the basis that one of the constituents lacked a property it could not possibly have. The property had been eliminated during the course of other revisions, and the effects of the revisions were not fully propagated. A similar oversight in one of the rules for the auxiliary "do" cost a great deal of unnecessary work during parsing by failing to reject when it should. This doubtless contributed to the failure of some sentences to parse because of limited resources. Errors of this kind are not always easy to detect and not always easy to correct.

Several failures involved "time" expressions. Developing an adequate grammar for time expressions is non-trivial because the grammar of time is extremely idiosyncratic. For instance, one may say "I work nights", where "nights" is a plural noun phrase, but not "I work night". Obviously, to allow noun phrases in general to serve adverbial functions would be explosive, and every effort was made to constrain the time adverbials. Some failures were due to overconstraining existing rules, some to non-existence of necessary rules. The lack of rules for dates caused the grammar to fail on two sentences that included them ("just after June 18th" and "since creation June 18"). It is not even clear whether the second expression ought to be regarded as grammatical.

A different kind of failure involved evaluation of alternative analyses, in which the sentence was parsed correctly, but the correct parse was not among those that were scored as being preferable on the basis of syntactic likelihood. This kind of failure is the subtlest and the trickiest of all to correct; correction will entail further tuning, extending, and testing. Too little is known about the distinctions that can and should be drawn between syntactically likely and syntactically possible combinations of words. However, according to Moore's figures, failure to evaluate correctly was surprisingly

infrequent. He lists only three cases for all eighteen dialogues. The next section contains a description of some of the strategies that were pursued in investigating this problem and building the distinctions into the rules.

3. Procedures, problems and policies

Broadly speaking, the changes in the grammar fell into one of three categories. The types of change, together with the procedures for making them and some of the related problems, were as follows:

Purging the grammar of elements specific to the old domain.

Process:

- Deleting words and categories.
- Eliminating their traces from the remainder of the grammar.

Problems:

- Disentangling rule interactions. In effect, all of the problems of extending the grammar, seen from an inverted perspective.
- Separating subgrammars for specialized domains and semantic fields from the general grammar.

Adding new vocabulary.

Process:

- Creating new categories for new words.
- Assigning "old" words to additional categories.
- Adding new attributes to lexical entries.

Problems:

- Discriminating between homographs and multiple word-senses.
- Discriminating between category differences and subcategory differences.

Folding the new vocabulary into the rules.

Process:

- Adding new phrase categories and rules.
- Adjusting old rules by adding or deleting constituents.
- Modifying the constructors.

Problems:

- Tracing ripple effects.
- Controlling rule interactions
to eliminate incorrect analyses and avoid
combinatorial explosions.

It should be emphasized that this analysis does not reflect different phases in which the grammar was gradually extended and adapted nor the order in which problems arose. Adding or subtracting a single new word to a grammar could conceivably invoke all of the problems mentioned above. The changes are categorized here to provide an idea of the strands that will be encountered in the following description of the intertwined processes and problems and of the strategies and policies for dealing with them.

3.1 Purging a grammar

As a preliminary to the additions and modifications designed to extend the grammar to the new domain, there was an initial cleanup to remedy some known defects. There was also a purge of some domain-dependent categories (e.g., rules for parsing names and titles of naval officers) and of the rules in which those categories appeared. In sum, 7 categories and 14 rules were deleted, almost balancing the additions. Because of DIAGRAM's initial limitations, it was relatively easy to decide what parts were so highly specialized that they would be used only in the domain in which it had been tested, or in some closely related domain.

It is worth noting, in this connection, that any sustained attempt to extend and adapt a grammar to a domain is likely to entail the development of specialized categories that must be folded into the rules, in order to handle the specialized language conventions developed by those who habitually communicate with each other about the objects and events in that domain. These categories and the rules in which they appear can be thought of as a subgrammar for the domain. However, they do not constitute a subset of the categories and rules of the grammar that contains them, at least not in the sense that they can always be easily extracted.

Ideally, they could be permitted to accumulate in a grammar that was being extended to represent the language exhaustively. Practically, an exhaustive grammar is not feasible because of the problem of categorial and combinatorial explosions. One reason is that we do not know enough about the semantic and pragmatic constraints necessary to prevent exponential explosions in the number of parsings that would result from having so many rules and categories in such a general grammar. We also do not know how to represent formally the knowledge that we do have, as native speakers, about such constraints. What we need is formalized knowledge and programmable procedures for representing when and how to "shift gears" from the language suitable for one discourse situation to another, including shifts in vocabulary, style (degrees of formality, etc.), and modes of referring. Until we have advanced further in such directions, it seems better to aim for a general or "core" grammar, rather than an exhaustive one. Such a grammar should contain the common words that are likely to be used in any discourse about any domain, so that it can be adapted with relative ease to special cases by adding to and tuning its vocabulary and rules. The words should be assigned to their usual categories with their usual non-specialized meanings.

Admittedly, it is not clear where to draw the line between generality and specialization. Even in a general grammar there will be some words and rules that are rather like subgrammars for a particular semantic field. The most significant of these are the words and rules that define the syntax for expressions referring to time and place. One might add to this, expressions of manner and purpose, from which it will appear that we are talking in semantic terms about a subgrammar for adverbial expressions. This is not such an odd way of thinking about adverbials as one might think. Adverbs and adverbial expressions are not a coherent syntactic class to the degree that noun phrases, verb phrases, and adjective phrases are. They include not only prepositional phrases like "on Wednesday", but also nouns and noun phrases like "today" and "last Friday" in the expressions for time, and words like "probably" and "possibly", derived from adjectives, in expressions of propositional attitudes. Their syntactic distribution is strongly affected by their semantic properties.

While the original DIAGRAM did contain rules for prepositional phrases, the rules for adverbial expressions of time were minimal. This posed no problem for the original domain, which was a static data base. None of the questions in the original domain contained prepositional phrases denoting locations in time. Although some of them did ask when a ship would arrive, the grammar was tested on answers in the form of "the ETA (Expected Time of Arrival) is...." In the new domain, on the other hand, expressions like "on Wednesday", and "last Friday" were common.

Another semantic field in which the extended grammar differed from the original is the field of mental states; that is, the field associated with words like "think", "know", "want", "guess",

"wonder", and "need". The semantic field associated with verbs of communicating, like "tell" and "say" is similar. All of these words, as verbs, combine with quite complex sets of complements, compared to verbs like "command", "reach", and "carry" (in their central senses). The original grammar defined the syntax for some of the subgrammars of mental states and of communication, but since there were no translators for their interpretation, they were not well tested. The new domain abounded in such words. The first four dialogues alone had two occurrences of "know", two of "think" and four of "want".

Although I have been referring to the parts of the grammar needed to analyze expressions in these semantic fields of time, place, mental states, and communication as "subgrammars," they are obviously more central and more general than the parts needed for handling ship names and officer titles. Purging them from the grammar would also be more difficult. One of the benefits of the exercise of extending DIAGRAM was to make it a more general core grammar for future extensions in any direction.

3.2 Adding vocabulary

In extending a grammar to cover a new domain, it is obviously necessary to add to the lexicon any new words that are needed for carrying on a discourse about the domain. The obvious first step in adding words is to assign them to a syntactic category. Most, but not all of them, will be assigned to a so-called "lexical" parts-of-speech category, although some words have syntactic properties that make assignment to phrasal categories. A concordance of sample texts, showing the context for each word token, is imperative for assignment of specialized words that have not found their way yet into any dictionary. How else, for example, are the lexical entries for "mtacpy" and "sndmsg" to be determined? A concordance is needed even for words that are in a dictionary, because the contexts contain clues for syntactic attributes of words that are central to the domain, and that are therefore likely to be used in unfamiliar, specialized ways.

Contexts are also needed to reveal new syntactic possibilities for a word already in the lexicon, indicating that its lexical entry needs revising or that it requires additional entries as a homograph. For example, "interrogate" is used in the Helper dialogues as a noun. Moreover, the contexts show that it has the syntactic properties of a proper noun. It is the name of a program. A standard dictionary entry (cf. Webster's Seventh New Collegiate Dictionary) shows it only as a verb.

Another example of a new word to be added to DIAGRAM's lexicon was "terminal". It also is a homograph, and is assigned by lexicographers ordinarily to the major word categories, noun and adjective. In a domain in which the "speaker" and "hearer" are using computer terminals, as they

are in the Helper domain, its most likely use is as a noun. Inspection of the concordance (a KWIC index) shows only two tokens, and in both occurrences it is indeed a noun, as expected. This example differs from the previous one in that "interrogate" was used in a new syntactic category, while "terminal" is used in one of its customary categories but with a specialized meaning.

In the preceding section, I spoke of purging the grammar of its specialized vocabulary when extending and adapting it to new domains. It should be clear by now that this is not just a matter of deleting some words from the lexicon. There are ways of tuning and untuning the vocabulary in a grammar like DIAGRAM without adding or excising words, but by weighting their lexical entries. Thus the two entries for "terminal" could be weighted for the Helper domain so that the noun assignment is preferred. The weighting would then be used to guide the parsing of sentences containing that word to prefer an analysis with that assignment in cases where alternatives are possible.

DIAGRAM currently employs a device called "factors" to assign likelihood scores to phrases that are constructed by the rules, but factors have not, so far, been used to assign weights to homographs. As the grammar grows and as more information is accumulated on what the likelihoods actually are for a given domain, the employment of such weights might usefully be explored. On the practical side, they might provide controls on multiple analyses; on the theoretical side, they might play a role in a model of how people are able to handle homography so easily. In this experiment, however, words like "terminal" were entered only in their most likely category if their other possible assignments were distinctly less likely. For example, "well" was entered only as an adverb, although in other domains one might speak of "a well" or of water "welling up". One very practical reason for not attempting the more ambitious treatment was lack of storage capacity for the intermediate parsing products that would occur if all possible combinations were being considered.

In addition to the problem of homography, there is a closely related problem of multiple word senses within a single categorial assignment. The word "have" is a striking example. It is treated as a homograph belonging to two distinct syntactic categories, as an auxiliary (HAVE) and as a verb (V). As a V, it combines with different complements to reflect two quite different senses. In "if you have any trouble...", it combines with a NP object, in the sense of (roughly) POSSESS NP; in "have him call me", it combines with an NP object and a VP, with a semantic translation (roughly) of CAUSE NP to VP.

Subtler and more numerous are differences in word senses that are correlated with the different particles and prepositions that appear in complements. Thus, the N/V homograph "look" has one sense as a verb when it combines with the preposition "at" and another when it combines with the particle "up". Both combinations share many very similar syntactic configurations, as in:

I was just looking at Groupstat (cf. inspect)
 I was just looking up the number of the reel. (cf. seek)

Both occur in passives, as in:

It was carefully looked at.
 It was carefully looked up.

However, "He quickly looked it at" is ungrammatical, while "He quickly looked it up" is well-formed.

Moreover, when there is a pronominal object in the active voice, the sentences

• He looked it at
 and • He looked up it

are ungrammatical, while

He looked at it
 and He looked it up

are well-formed.

It might be possible to handle multiple senses of words by multiplying the number of lexical categories in the grammar to permit assigning a word to as many categories as it had differing senses that were correlated in some way with different syntactic effects. There would be categories like VTRANSITIVE, VINTRANSITIVE, VCAUSATIVE, NTIME, NHUMAN, etc. This policy would add so many highly specialized categories to a grammar that it would multiply the rules enormously. Such a solution is clearly neither theoretically elegant nor computationally feasible.

It also proved not feasible to have more than one entry for a word within a given category, where each entry is coded with different properties for each differing sense the word has within that category. (This is essentially the way standard dictionaries enter different senses of nouns, verbs, and adjectives.) In fact, an initial decision to have an entry for each sense of the verb "have" proved disastrous, computationally, within the resource limitations. The decision therefore was to limit the categories for content words to the traditional ones and to make only one entry for a word within a lexical category. Semantics was left with the task of discriminating the senses according to the syntactic and semantic context. This is a reasonable tactic, but in many cases it did result in some incorrect analyses along with correct ones when rule interactions allowed both the "right" word-sense interpretations and the "wrong" ones.

3.3 Folding the new vocabulary into the rules

As pointed out in the section on additions and modifications, relatively few new rules were added to DIAGRAM. Many rules were modified to accept new constituents, especially adverbs, but the major accommodations to the new vocabulary and the new style of communicating that the domain introduced were made by modifying the constructors.

Deciding how to modify rules to accommodate new words and categories is much like deciding what lexical categories to assign to a word. That assignment requires knowledge of the environment in which the word is used in the domain. The same KWIC index of the dialogues that supplied data for making the category assignments also showed what complements and modifiers could accompany a word. In the codes for a word's lexical entry as, say, a verb, it was assigned properties according to the complements it could acquire as head of a verb phrase. Where the complements were particles or prepositional phrases, the actual form of the particle or introductory preposition was given as the value of an attribute. Two verb entries below illustrate the coding.

```
CHECK  (DIROBJ . T) (PREPCOMP WITH FOR)(PARTICLE OUT))
GET    (DIROBJ . T)(INDIROBJ . T)(PREDOBJ . T)(PARTICLE BACK)
      (PREPCOMP TO))
```

This coding was exploited heavily in the constructors to constrain the attachments of prepositional phrases to verbs and nouns. It was possible to reject some combinations and raise the score for others in applying a rule for attaching a PP when that PP was introduced by a preposition on the list of P's known to be tightly combinable with the word (V, N, or A) that actually occurred in the sentence. For example, there are many possible parses for the verb phrase "send a file to the printer from FTP", but since the lexical entry for "send" shows that it is especially likely to combine with "to" and "from", the analysis that attaches "to the printer" and "from FTP" to the VP as adverbial modifiers will be preferred over analyses that attach them as modifiers of "file" ("a file to the printer from FTP") or "printer" ("the printer from FTP").

As a matter of policy, we did not add the full range of possibilities for the prepositions that could be tightly combined with NVAs, but only those that actually occurred with them in the texts. There were several motives for this decision. One was caution. It was not clear whether adding the information would decrease or increase combinatorial explosions in prepositional attachments. Another was an inclination to follow the lead of the actual dialogues. It seemed better to build the extension to the domain in a controlled way rather than resort to intuitions or exhaustive dictionary entries. As a result, it is often easy to think of additional prepositions and particles that could be added to the entries for verbs like "check" and "get" shown above. It also means that the next round of tests over additional dialogues may fail to pick the preferred analyses for some sentences.

The technique just described was extremely helpful in reducing the problem of multiple analyses of phrases containing particles and prepositions or, more generally, adverbial expressions and postnominal modifiers. This is not to say that the problem is solved and that other techniques will not

be necessary. What can be claimed is that further explorations along these lines should help build into a grammar a great deal of syntactic/semantic information to constrain its analyses.

4. Summing up and pointing ahead

There is always a great deal to be learned in testing a formal grammar in a new domain. The exercise is not only good for debugging but also for pointing out essential limitations and hitherto unnoticed opportunities. Any further extension of DIAGRAM would follow much the same course as this one. There would be a preliminary cleanup of the grammar and a re-shaping of the rules in the light of the experience gained so far. In particular, I would advocate reworking the rules so that multiple word-senses were keyed to the rule that combined a word with its complements. Thus "have" would be combined with its various complements by quite distinct rules, rather than by a rule schema that captures the various optional constituents. This would not work for different word-senses that were correlated with differences in the actual prepositions themselves rather than differences in the syntactic types of complements. That is, the rule that combines "have" with a single NP would be different from the rule that combines it with an NP followed by an infinitive expression, but the same rule that combines "look" with "at NP" should also combine it with "for NP".

But these are decisions that involve large issues of syntactic theory and of computational efficiency. It is important to keep the issues separate in an analysis, but it is illuminating to bring them together in an exercise of this sort, to study the consequences of decisions made in either area when a grammar is tested as thoroughly as this one was.

I. The four dialogues used for developing the grammar

	<u>SUCCESSFUL</u>	<u>UNSUCCESSFUL</u>	<u>TRIVIAL</u>
OC286.DIALOGUE⁴			
(COULD I POSSIBLY RETRIEVE THE FOLLOWING TWO FILES .)	X		
(I THINK THEY WERE ON OUR DIRECTORY LAST NIGHT .)	X		
(FILENAME1 .)			X
(FILENAME2 .)			X
(WHAT TAPE NUMBERS ARE THEY ?)	X		
(WASN'T A SYSTEM DUMP TAKEN LAST NIGHT ?)	X		
(YES .)			X
(I DON'T UNDERSTAND YOUR QUESTIONS THEN .)	X		
(DID YOU DELETE A FILE ACCIDENTALLY YESTERDAY AND YOU WANT IT BACK ?)	X		
(NO , I JUST NOW DELETED A FILE .)	X		
(AND YOU WANT IT ?)			
(CORRECT .)			X
(I'LL SEE IF I CAN RESTORE IT , HOWEVER I DON'T KNOW HOW SOON I CAN DO IT .)	X		
(THE MAGTAPE UNITS ARE UNAVAILABLE RIGHT NOW .)	X		
(TIME IS NO PROBLEM .)	X		
(I NEED THOSE TWO .)	X		
OC30.DIALOGUE			
(CAN I RECOVER FILENAME FROM THE MOST RECENT SYSTEM DUMP ?)	X		
(PROBABLY .)			X
(LET ME LOOK FOR IT AND GET BACK TO YOU , OK ?)	X		
(COULD YOU SENDMSG TO ME ABOUT IT ?)	X		
(IS THERE ONLY ONE VERSION NUMBER , SO I WON'T HAVE ANY TROUBLE DECIDING WHICH ONE TO RESTORE ?)	X		
(THERE SHOULD BE ONLY ONE .)	X		
(PICK THE HIGHEST VERSION , IF THERE ARE MORE THAN ONE .)	X		
(DO YOU KNOW IF IT WAS THE MOST RECENT FULL OR INCREMENTAL DUMP ?)	X		
(WHEN WERE THEY TAKEN ?)	X		
(I TOOK AN INCREMENTAL DUMP TONIGHT .)			
(THE LAST FULL DUMP WAS TAKEN LAST FRIDAY NIGHT .)	X		
(CHOOSE THE LAST INCREMENTAL DUMP WHICH WAS TAKEN MORE THAN TWENTY MINUTES AGO .)	X		

⁴These dialogues were very slightly regularized before being used in these experiments: spelling was corrected to standard English spelling, and in cases where informal judgment indicated that ellipsis had occurred, with an obvious restoration, the restoration was made. The marks for success and failure are principally, but not entirely, from the same runs which were used for computing percentages of success. Where a sentence is not marked, it usually indicates occurrence of a linguistically irrelevant computational problem.

SUCCESSFUL UNSUCCESSFUL TRIVIAL

OC405.DIALOGUE

(CAN I GET TAPE1 ON A DECTAPE DRIVE ?)	X		
(I WILL HAVE TO SWITCH THE DECTAPES OVER TO THIS MACHINE .)	X		
(YOU DON'T HAVE ACCESS TO DEVTENEX , DO YOU ?)	X		
(NO .)			X
(I'M JUST A LOWLY NET USER .)	X		
(BUT , I AM TOLD			
THAT SWITCHING THE DRIVES TENDS TO CAUSE CRASHES .)	X		
(IT IS NOT WORTH IT .)	X		
(THAT HAPPENS ONLY IF			
SOMEBODY SCREWS UP HERE AND PUSHES THE WRONG BUTTON .)	X		
(DO YOU STILL WANT IT ?)	X		
(OK .)			X
(THE SWITCH IS MADE .)	X		
(NOW FOR TAPE1 .)		X	
(DO YOU WANT IT WRITE ENABLE ?)	X		
(NO .)			X
(ALSO , CAN I GET TAPE2 WITHOUT WRITE ENABLE ?)	X		
(OK .)			X
(IS ARCHIVE BEING RUN MORE THAN ONCE PER WEEK , YET ?)	X		
(NO .)			X
(OK , TAPE1 IS ON DTA1 WITHOUT WRITE ENABLE			
AND TAPE2 IS ON DTA2 WITHOUT WRITE ENABLE .)	X		

OC8.DIALOGUE

(COULD YOU MOUNT A MAGTAPE FOR ME !)	X		
(IT'S TAPE1 !)	X		
(NO RING PLEASE !)		X	
(CAN YOU DO IT IN FIVE MINUTES !)	X		
(WE ARE NOT ALLOWED TO MOUNT THAT MAG TAPE .)	X		
(YOU WILL HAVE TO TALK TO			
OPERATOR ABOUT IT , AFTER NINE A.M MONDAY THROUGH FRIDAY .)		X	
(HOW ABOUT TAPE TAPE2 ?)	X		
(NO .)			X
(GO AHEAD .)	X		
(I AM NOT EXACLY SURE OF THE REASON			
BUT WE WERE GIVEN A LIST OF USERS WE ARE			
NOT SUPPOSED TO MOUNT MAGTAPES FOR AND LINKER IS ON IT .)		X	
(I THOUGHT WE COULD DO IT AT NIGHT .)	X		
(IS THERE ANY TIME PERIOD ON THAT LIST ?)	X		
(NO .)			X
(OK .)			X
(YOU MIGHT CHECK WITH NAME1 .)	X		
(PERHAPS THERE			
IS SUPPOSED TO BE A TIME LIMIT , AND IT WAS FORGOTTEN .)	X		
(YES .)			X
(I'LL DO THAT .)	X		

II. The fourteen additional dialogues used for testing the grammar

SUCCESSFUL UNSUCCESSFUL TRIVIAL

OC188.DIALOGUE

(DO YOU HAPPEN TO KNOW HOW TO ROUTE A FILE TO THE PRINTER FROM FTP ?)		X	
(IS THAT DIFFERENT ?)	X		
(FROM WHAT ?)		X	
(FROM NORMAL .)		X	
(YES , YOU USE THE LIST COMMAND NORMALLY .)	X		
(BUT THAT'S NOT POSSIBLE WITH FTP .)	X		
(I'LL SEE IF I CAN LOOK IT UP , OK ?)	X		
(OK .)			X
(I CAN'T FIND THAT INFO .)	X		
(HOW URGENT IS IT ?)	X		
(NOT VERY .)	X		
(I'LL WAIT TILL I CAN FIND SOMEONE WHO KNOWS .)		X	
(OK .)			X

OC193.DIALOGUE

(DO YOU KNOW WHO CAN GIVE ME INFO ABOUT THE LONDON TIP ?)		X	
(WHAT KIND OF INFO ?)		X	
(WELL , WHERE IT'S LOCATED, WOULD A TERMINAL BE AVAILABLE FOR A TRAVELER , WHO TO CONTACT IN LONDON ONCE THERE , ETC .)		X	
(IN OTHER WORDS , YOU WOULD LIKE TO TALK TO OR HEAR FROM SOMEONE IN LONDON OVER THE NET .)		X	
(IS THAT CORRECT ?)	X		
(THAT WOULD BE BEST .)	X		
(HOWEVER , I HAVE BEEN UNABLE TO RAISE THE OPERATOR AT UK-ICS TO FIND OUT ANYTHING DIRECTLY .)		X	
(I WAS TOLD THAT THERE WAS SOMEONE AT ISI WHO WAS A PUBLIC RELATIONS MAN FOR THE TIP ?)		X	
(CAN YOU HOLD WHILE I CHECK THAT OUT ?)	X		
(YES .)			X
(SHOULD WE KEEP THE LINK ?)	X		
(IT IS BETTER IF I LINK BACK , OK ?)	X		
(OK .)			X

SUCCESSFUL UNSUCCESSFUL TRIVIAL

OC223.DIALOGUE

(ON WEDNESDAY , I CREATED A RATHER			
LENGTHY LISTING ON THE LINE PRINTER .)	X		
(I HOPE IT HASN'T BEEN DISCARDED .)	X		
(I NEGLECTED TO ASK ANYONE TO HOLD IT ?)	X		
(IT HAS BEEN DISCARDED .)	X		
(OKAY , I'LL DO IT AGAIN .)	X		
(DON'T THROW IT AWAY .)	X		
(I'LL PICK IT UP THIS AFTERNOON .)	X		
(OK , I'LL SEE YOU THEN .)	X		
(DO YOU KNOW WHERE TO COME ?)	X		
(YES .)			X
(I'VE BEEN THERE BEFORE .)	X		
(OK .)			X

OC235.DIALOGUE

(I HAVE JUST DISCOVERED THAT THIS FILE IS TRUNCATED .)	X		
(I WANT TO FIND EITHER ANOTHER COPY			
OF THIS VERSION , OR A PREVIOUS VERSION .)		X	
(CAN YOU HELP ME ?)	X		
(I'LL LOOK THROUGH THE DUMP LISTING .)	X		
(WHAT IS THE MOST LOGICAL DUMP IT WOULD BE IN ?)		X	
(THE FULL DUMP JUST AFTER JUNE NUMBER1 .)		X	
(IS IT STILL AROUND ?)	X		
(IT WOULD BE IN NONE			
AFTER THAT LIKE LAST FRIDAY'S FULL DUMP .)		X	
(WHAT ?)			X
(DID YOU HAVE IT IN YOUR DIRECTORY LAST FRIDAY ?)	X		
(YES .)			X
(IT HAS BEEN THERE CONTINUOUSLY SINCE CREATION JUNE NUMBER1 .)		X	
(BUT AT SOME TIME IT GOT MESSED UP .)		X	
(OK I'LL START LOOKING THROUGH THE DUMPS .)		X	
(HOW LONG IS THE CORRECT ONE ?)		X	
(IT IS IN THAT SAME RANGE .)	X		
(IT IS NO MORE THAN NUMBER2 PAGES LONG .)		X	
(MEANWHILE , I WILL LOOK IN			
THE ARCHIVE LISTINGS FOR A BACKUP OR EARLY VERSIONS .)		X	
(I'LL LINK AGAIN IF I FIND ANYTHING .)	X		
(OK .)			X

SUCCESSFUL UNSUCCESSFUL TRIVIAL

OC253.DIALOGUE

(I DON'T KNOW IF YOU KNOW , BUT OCCASIONALLY WE LOG IN THERE AND ASK YOU TO MOUNT A TAPE FOR US .)	X		
(WE THEN READ A FILE ONTO IT AND SOMEONE PICKS IT UP IN THE MORNING .)	X		
(WHAT TAPE NUMBER RANGE ARE YOU SPEAKING OF ?)		X	
(I'M NOT SURE I KNOW WHAT YOU MEAN .)		X	
(WE USUALLY GET IT ON DRIVE0 .)	X		
(I AM GOING TO RUN MTACPY .)	X		
(WE GAVE YOU A BUNCH OF TAPES A LONG TIME AGO .)		X	
(OK , I'LL REPHRASE .)	X		
(WHAT TAPE NUMBER DO YOU WANT ME TO MOUNT ?)	X		
(DO YOU MEAN THE NUMBER OF THE REEL ?)	X		
(YES .)			X
(I DON'T KNOW .)	X		
(THEY DIDN'T HAVE NUMBERS WHEN WE SENT THEM .)			
(I AM GOING TO PUT A FILE ONTO THE TAPE FROM A FILE THAT I HAVE ONLINE THERE SO IT DOESN'T MATTER FROM THAT ANGLE .)			
(OK , I WILL SEE IF I CAN LOCATE YOUR TAPES .)			
(I WILL LINK BACK TO YOU SHORTLY , OK ?)			
(GREAT .)			X

OC396.DIALOGUE

(I ASKED NAME1 TO SET UP AN ACCOUNT FOR NAME2 WITH PASSWORD WORD1 .)	X		
(HE SENT ME A NOTE THAT HE DID SO BUT I COULDN'T LOGIN TO THE NEW ACCOUNT AND NEITHER COULD NAME2 .)		X	
(WHAT IS WRONG ?)	X		
(NAME1 WILL BE HERE SOON .)	X		
(CAN YOU LINK BACK IN A FEW MINUTES ?)	X		
(NO, WHY CAN'T YOU FIX IT OR CHECK NOW ?)	X		
(I CAN'T DO THAT .)	X		
(ONLY NAME1 CAN DO IT .)	X		
(OK, HAVE HIM CALL ME .)	X		

OC497.DIALOGUE

(I'D LIKE TO PUT SOME OF OUR FILES ON TAPE .)	X		
(ARE YOU FAMILIAR WITH THE PROCEDURES ?)	X		
(NO , BUT I AM MORE THAN WILLING TO HELP .)		X	
(BUT THERE IS A PROBLEM .)	X		
(I AM AT PRESENT RUNNING A FULL DUMP AND IT WILL BE A FEW HOURS BEFORE I CAN USE THE TAPE DRIVES .)		X	
(I CAN USE THE DECTAPE DRIVES .)	X		
(NO , THE DEC DRIVES WON'T DO WHAT I NEED .)	X		
(I'LL BE BACK IN THE MORNING .)	X		
(OK , I HOPE THAT THIS WASN'T TOO MUCH OF AN INCONVENIENCE TO YOU .)	X		
(NO , THAT'S OK .)	X		
(ONE TAKES ONES CHANCES AT A TIME LIKE THIS ANYWAY .)	X		

SUCCESSFUL UNSUCCESSFUL TRIVIAL

OC549.DIALOGUE

(I WAS JUST LOOKING AT GROUPSTAT AND NOTICE THAT THERE ARE SOME DETACHED ACCOUNTS WITH NUMBER1 HOURS PILED UP .)			X
(IF I GET DETACHED , DOES THE SYSTEM THROW ME OUT AFTER AWHILE ?)			X
(NO , NOT TO MY KNOWLEDGE .)			X
(THE ONLY WAY FROM WHAT I UNDERSTAND THAT YOU WILL LOSE THAT DETACHED JOB IS IF THE SYSTEM HAPPENS TO CRASH WHILE YOUR JOB IS DETACHED .)			X
(OK , THAT EXPLAINS THE DETACHED JOBS WITH MANY HOURS PILED ON IT .)	X		
(I HAVE BEEN TELLING GUYS HERE THAT I THOUGHT THE SYSTEM DID THROW YOU OUT , SO I GUESS I WILL HAVE TO CORRECT THAT .)			
(WELL , IT WAS A MISUNDERSTANDING .)			
(WAIT , BEFORE YOU START CORRECTING PEOPLE LET ME CHECK TO BE SURE THAT I AM UNDERSTANDING IT CORRECTLY .)			
(BECAUSE I WOULDN'T WANT TO LEAD YOU WRONG EITHER .)			
(I JUST DON'T KNOW IT FOR A FACT AND I WOULD LIKE TO GET A BACKUP FROM SOMEONE WHO WOULD KNOW WITHOUT A DOUBT .)			
(WHAT I WILL DO IS CHECK ON IT AND SEND YOU A MESSAGE OR LINK TO YOU LATER ON TODAY OR FIRST THING IN THE MORNING .)			
(SO , HOLD ON FOR A WHILE , OK ?)			
(OK .)			X

OC81.DIALOGUE

(DO YOU KNOW IF THE SYSTEM WILL REALLY BE UP ALL NIGHT ?)
(UNLESS WE CRASH !)

OC92.DIALOGUE

(YOU LINKED EARLIER .)			X	
(YES .)				X
(WE ARE INTERESTED IN GETTING A DISK FILE OF OURS TRANSFERED TO A MAGTAPE SO THAT WE CAN HAVE THE SOURCE PUNCHED ON CARDS .)			X	
(DO YOU SUPPOSE THAT WE COULD SEND A TAPE TO YOU AT LOC1 AND GET IT MOUNTED SOME TIME SO THAT WE CAN USE BCDTAP TO COPY IT ?)			X	
(HOW MUCH TIME WILL YOU WANT TO USE THE MAGTAPE UNITS ?)	X			
(IT WOULD PROBABLY ONLY TAKE ABOUT NUMBER1 MINUTES .)			X	
(OK , SEND IT TO NAME1 AT THE LOC1 ADDRESS .)	X			
(DO YOU HAVE IT ?)	X			
(NO .)				X
(IT'S ADDRESS1 , OK ?)	X			
(OK .)				X
(WE WILL PROBABLY SEND THE TAPE NEXT WEEK SOME TIME .)			X	
(SHALL WE JUST ASK HER TO HOLD ON TO THE TAPE OR WHAT ?)			X	
(YES .)				X
(WHEN YOU SEND IT , SNDMSG TO NAME1 TELLING HER TO EXPECT IT AND WHAT YOU WANT DONE WITH IT WHEN IT ARRIVES , OK ?)			X	
(OK .)				X

SUCCESSFUL UNSUCCESSFUL TRIVIAL

OC613.DIALOGUE

(HOW LONG DOES IT TAKE TO GET SOMETHING UNARCHIVED ?)	X		
(OUR TAPE UNITS WERE DOWN AND WE ARE JUST NOW TESTING THEM TO SEE IF THEY WILL WORK .)		X	
(IT WAS A GENERAL TYPE QUESTION .)	X		
(I'VE TRIED BEFORE , MAYBE THREE TIMES , TO GET THIS ONE FILE BACK WITHOUT ANY LUCK .)		X	
(AM I DOING SOMETHING WRONG ?)	X		
(I USED INTERROGATE AND GOT IT TO ASK ME IF I WANTED IT RETRIEVED , BUT NOTHING EVER SHOWED UP .)		X	
(IF YOU USED THE INTERROGATE COMMAND IT SHOULD HAVE SENT ME A REQUEST FOR THE TAPE .)	X		
(SINCE YOU SAY IT NEVER WAS RESTORED I MUST ASSUME THAT THERE IS SOMETHING WRONG WITH THE INTERROGATE COMMAND AS IT EXISTS IN THIS SYSTEM AT PRESENT .)		X	
(I WILL TELL THE PROPER AUTHORITIES ABOUT IT .)	X		
(IN THE MEANTIME IF YOU WILL TELL ME THE FILE YOU WANT RESTORED I WILL RESTORE IT FOR YOU IN A FEW MINUTES , OK ?)		X	
(FINE .)			X
(I JUST WANTED TO MAKE SURE I WASN'T MISSING SOME CRUCIAL STEP IN THE PROCEDURE FOR GETTING IT BACK .)		X	
(ALSO , I ASKED FOR TWO FILES BACK JUST A COUPLE OF MINUTES AGO , SO YOU MIGHT SEE IF YOU GOT ANY MESSAGES ABOUT THEM .)		X	
(ANYWAY , I WOULD LIKE FILENAME1 AND FILENAME2 RESTORED .)	X		
(OK , WHO'S FILES ARE THESE ?)	X		
(LINKERNAME .)			X
(OK , IF YOU WILL STAY LINKED FOR A FEW MINUTES LONGER I WILL CHECK FOR THE MESSAGES .)		X	
(OK .)			X
(THE FILES ARE HERE AND HAVE BEEN REQUESTED AS THE SUBPROGRAM IS SUPPOSED TO DO .)		X	
(AS I SAID EARLIER , THE TAPE UNITS WERE DOWN SO I HAD NOT CHECKED THE ANYREQ FILE FOR A FEW MINUTES .)		X	
(I WILL RESTORE THEM FOR YOU IN JUST A FEW MINUTES , OK ?)	X		
(THAT WILL BE JUST FINE .)	X		
(I WAS BEGINNING TO WONDER WHAT WAS GOING ON AFTER SEVERAL UNSUCCESSFUL TRIES AT GETTING THEM BACK .)		X	
(THAT'S OK .)	X		
(THAT'S WHAT I AM HERE FOR .)		X	
(IF YOU HAVE ANY TROUBLE IN THE FUTURE FEEL FREE TO ASK ANY OPERATOR FOR HELP .)	X		

OC624.DIALOGUE

(I WOULD LIKE YOU TO RETRIEVE THE ENTIRE DIRECTORYNAME DIRECTORY FOR FEBRUARY .)	X		
(IT'S ON TAPES NUMBER1 OR NUMBER2 .)		X	
(OK , IT WILL TAKE A LITTLE WHILE TO RESTORE ALL OF THE FILES .)	X		
(DO YOU KNOW HOW TO DO IT WITHOUT GIVING EACH NAME INDIVIDUALLY ?)	X		
(NO , I WAS GOING TO LOOK IT UP THOUGH OR ASK YOU .)	X		
(I DON'T REALLY REMEMBER BUT THERE IS A WAY .)	X		
(IF YOU CAN'T FIND IT LINK BACK .)	X		
(OK , I WILL START RETRIEVING YOUR FILES IN A FEW MINUTES .)	X		
(OK .)			X

SUCCESSFUL UNSUCCESSFUL TRIVIAL

OC66.DIALOGUE

(THIS IS NAME1 , I ASSUME ?)	X		
(WRONG , THIS IS NAME2 .)	X		
(OK , IT DOES NOT MATTER .)	X		
(IS THERE PAPER TAPE IN THE PUNCH ?)	X		
(I WOULD LIKE TO PUNCH SOME STUFF AND GET IT OFF THE DISK .)	X		
(THERE IS PAPER IN THERE .)	X		
(DO I NEED TO TURN ANYTHING ON , OR DO YOU TAKE CARE OF IT ?)	X		
(IT SHOULD BE ON ALREADY , I THINK .)	X		
(LET ME TRY IT , OK ?)	X		
(THE POWER LIGHT IS NOT ON .)	X		
(CAN YOU TURN ON THE POWER SWITCH .)	X		
(I CAN ASSIGN THE DEVICE TO MYSELF			
BUT I CANNOT TURN ON THE POWER FROM THE TERMINAL !)		X	
(I DON'T KNOW HOW TO TURN ON THE POWER .)	X		
(IF YOU CAN GIVE ME A FEW MINUTES , I CAN			
TRY TO FIND OUT AND THEN YOU TRY TO START , OK ?)	X		
(OK , I THINK NAME1 KNOWS .)	X		
(HE IS ON NOW .)	X		
(I WILL BREAK AND YOU CAN GET BACK WHEN YOU ARE READY , OK ?)	X		
(OK .)			X

OC75.DIALOGUE

(THIS IS NAME1 .)	X		
(I WOULD LIKE TO DE-ARCHIVE DIRNAME			
VIA INTERROGATE-STAR.STAR PREPARATORY			
TO MOVING THE ENTIRE DIRECTORY FROM LOC1 TO LOC2 .)	X		
(IS THIS PERMITTED ?)	X		
(HOW MANY FILES ?)		X	
(DO YOU HAVE ANY IDEA ?)	X		
(THERE ARE ABOUT NUMBER1 .)		X	
(NOTE WE FREED ABOUT NUMBER2 YESTERDAY .)		X	
(THE PAGES ARE NO PROBLEM , BUT IT IS LIABLE			
TO TAKE FOREVER IF THEY ARE NOT ON VERY FEW TAPES .)		X	
(THERE ARE ABOUT NUMBER3 FILES INVOLVED ON THREE PAIRS OF TAPES .)		X	
(IS IT URGENT TO DO THIS NOW ?)		X	
(YES .)			X
(OK , FIRE AWAY AND WE WILL GET THEM DONE AS SOON AS POSSIBLE .)	X		

III. The final rules

The rules below are derived directly from the final computational rules, with some systematic adjustments for readability. The conventions are those used in [Robinson 82], which also describes the grammar formalism and content.

(GRAMMAR.DEFQ "10-FEB-81 10:21:10" ****frozen****)

(Category A)

(RulesUsing A are DETQ1 NP1 and NP2)

(Category ADJ)

(RulesUsing ADJ are ADJ1 ADJ2 ADJP1 ADJP2 ADJP3 ADJP6X ADV1 ASNP2
NOMHD2 NOMHD3 NPEST and WHADJP1)

(Rules for ADJ

(ADJ1 ADJ = ADJ # 1 CONJ ADJ # 2 ;)

(ADJ2 ADJ = ADV ADJ ;
CONSTRUCTOR

(AND (@ ADV ADJ)
(F.REJECT (QUOTE F.ADJ2))))
)

(Category ADJCOMP)

(RulesUsing ADJCOMP are ADJP1 ADJP3 ADJP4 ADJP6X and WHADJP1)

(Rule for ADJCOMP

(ADJCOMP ADJCOMP = {OF NP / (FOR NP) INFINITIVE} ;)
)

(Category ADJP

(RulesUsing ADJP are ADJP4 NCOMP1 PRED1 RESPONSE1 SWH3 and VP2)

(Rules for ADJP

(ADJP1 ADJP = ADJ (ADJCOMP) ;
CONSTRUCTOR

(PROGN (OR (@ ADJCOMP)
(@SET NOCOMP T)))
)

(ADJP2 ADJP = (DETQ) ("NO") ER ("MUCH") ADJ (THANCOMP) ;
CONSTRUCTOR

(PROGN (OR (NOT (@ DETQ))
(NEQ (@ NBR DETQ)
(QUOTE PL))
(F.REJECT (QUOTE F.NBRQ)))))

(ADJP3 ADJP = ADJ ENOUGH (ADJCOMP) ;
CONSTRUCTOR

(OR (@ ADJCOMP)
(@SET NOCOMP T)))

(ADJP4 ADJP = AS ADJP ({ASCOMP # 1 (ADJCOMP # 1) / ADJCOMP # 2 (ASCOMP # 2)})
;
CONSTRUCTOR

(OR (@ ASCOMP # 1)
(@ ASCOMP # 2)
(@SET NOCOMP T)))

(ADJP6X ADJP = EST ("MUCH") ADJ (ADJCOMP) ;
CONSTRUCTOR

(OR (@ ADJCOMP)
(@SET NOCOMP T)))
)

(Category ADV)

(RulesUsing ADV are ADJ2 ADV2 ADVNBR ADVP1 ADVP2 ADVP6X ADVV AS1 BEP1
DOP1 HAVEP1 INF1 ING1 MODAL MODP PP5 PPL1
PRED1 QPP1 SUBCONJ1 and WHPP2)

(Rules for ADV

(ADV1 ADV = ADJ "LY" ;
CONSTRUCTOR

(AND (@ CONJ ADJ)
(F.REJECT 'F.ADV1)))

(ADV2 ADV = ADV # 1 ADV # 2 ;
CONSTRUCTOR

(PROGN (AND (OR (@ CONJ ADV # 1)
(@ CONJ ADV # 2))
(F.REJECT 'F.ADV2CONJ))
(AND (@ ADV # 1 ADV # 1)
(F.REJECT 'F.ADV2))))
)

(Category ADVP)

(RulesUsing ADVP are ADVP4 and RESPONSE1)

(Rules for ADVP

(ADVP1 ADVP = ADV (ENOUGH) ;)

(ADVP2 ADVP = (DETQ) ER ("MUCH") ADV (THANCOMP) ;
CONSTRUCTOR

(PROGN (OR (NOT (@ DETQ))
(NEQ (@ NBR DETQ)
(QUOTE PL))

(F.REJECT (QUOTE F.NBRQ))))

(ADVP4 ADVP = AS ADVP (ASCOMP) ;)

(ADVP6X ADVP = EST ("MUCH") ADV ;)
)

(Category AS)

(RulesUsing AS are ADJP4 ADVP4 AS1 ASCOMP ASNP and ASNP2)

(Rule for AS

(AS1 AS = ADV AS ;
CONSTRUCTOR

(AND (@ ADV AS)
(F.REJECT 'F.AS1)))
)

(Category ASCOMP)

(RulesUsing ASCOMP are ADJP4 ADVP4 ASNP and ASNP2)

(Rule for ASCOMP

(ASCOMP ASCOMP = AS {NP / SDEC} ;)
)

(Category AUXB)

(RulesUsing AUXB are SB1 SREL2 STHERE1 SWH3 and TAGAUX)

(Rule for AUXB

(AUXB1 AUXB = (MODALP) (HAVEP PPL) BEP (BEING) ;
CONSTRUCTOR

(PROGN (COND
((@ MODALP)
(COND
((@ HAVEP)
(OR (@ INFINITIVE HAVEP)
(F.REJECT (QUOTE BMB)))
(@SET PERFECTIVE T)))
(OR (@ INFINITIVE BEP)
(F.REJECT (QUOTE F.BMB)))
(@SET MODALITY T))
((@ HAVEP)
(OR (@ INFINITIVE BEP)
(F.REJECT (QUOTE BHB)))
(@FROM HAVEP TENSE NBR)
(@SET PERFECTIVE T))
(T (OR (@ TENSE BEP)
(F.REJECT (QUOTE BB)))
(@FROM BEP TENSE NBR PERSON)))
(COND
((@ BEING)
(@SET PROGRESSIVE T))))
)

(Category AUXD)

(RulesUsing AUXD are SD1 SREL2 SWH2 TAGAUX and VPCONJ)

(Rule for AUXD

(AUXD1 AUXD = {DOP / (MODALP) (HAVEP PPL) (BE ING)} ;
CONSTRUCTOR

```
(PROGN (COND
  ((@ DOP)
    (@FROM DOP TENSE NBR SUFFIXONLY))
  ((@ MODALP)
    (COND
      ((@ HAVEP)
        (OR (@ INFINITIVE HAVEP)
          (F.REJECT (QUOTE DOMH)))
        (@SET PERFECTIVE T)))
      ((@ BE)
        (OR (@ INFINITIVE BE)
          (F.REJECT (QUOTE DOMB)))
        (@SET PROGRESSIVE T)))
    (@SET MODALITY T))
  ((@ HAVEP)
    (COND
      ((@ BE)
        (OR (@ INFINITIVE BE)
          (F.REJECT (QUOTE DOHB)))
        (@SET PROGRESSIVE T)))
      (@FROM HAVEP TENSE NBR)
      (@SET PERFECTIVE T))
    ((NOT (@ INFINITIVE BE))
      (@FROM BE TENSE NBR)
      (@SET PROGRESSIVE T))
    (T (F.REJECT (QUOTE DOB))))))
)
```

(Category BE)

(RulesUsing BE are AUXD1 BEEN BEING1 BEP1 GERUND and STHEREQ2)

(Category BEEN)

(RulesUsing BEEN are SBQ3 and STHEREQ3)

(Rule for BEEN

(BEEN BEEN = PPL BE ;
CONSTRUCTOR

```
(PROGN (OR (@ INFINITIVE BE)
  (F.REJECT (QUOTE F.INFBEEN))))
)
```

(Category BEING)

(RulesUsing BEING are AUXB1 INF1 SBQ1 SBQ2 and SBQ3)

(Rule for BEING

(BEING1 BEING = ING BE ;
CONSTRUCTOR

(PROGN (@SET PROGRESSIVE T)
(OR (@ INFINITIVE BE)
(F.REJECT (QUOTE F.BEING))))
)

(Category BEP)

(RulesUsing BEP are AUXB1 INF1 SBQ1 SBQ2 SDQ2 SDQ3 SDQ4 and STHEREQ1)

(Rule for BEP

(BEP1 BEP = (ADV) BE ;
CONSTRUCTOR

(PROGN (@FROM BE TENSE NBR PERSON INFINITIVE)))
)

(Category CONJ)

(RulesUsing CONJ are ADJ1 SX1 and VPCONJ)

(Category DDET
CONSTRUCTOR (@SET QUANT (@ DIAMOND.SPELLING)))

(RulesUsing DDET are GERUND NP1 NP2 and NPEST)

(Rules for DDET

(DDET1 DDET = DET ((NUMBER / QPP)) ;
CONSTRUCTOR

(PROGN (COND
((@ NUMBER)
(OR (AGREE NBR NUMBER DET)
(F.REJECT (QUOTE F.NBRNUM)))
(@FROM NUMBER NBR)
(@SET TYPE (QUOTE COMMON)))
((@ QPP)
(OR (AND (AGREE NBR QPP DET)
(EQ (@ NBR QPP)
(QUOTE PL)))
(F.REJECT (QUOTE F.NBRQPP)))
(@SET NBR (QUOTE PL)))
(T (@FROM DET NBR INCOMPLETE)))
(@SET DEF T))
)

(DDET2 DDET = NP "GEN" ;
CONSTRUCTOR

(PROGN (@FROM NP DEF)))
)

(Category DET
)

(RuleUsing DET is DDET1)

(Category DETERQ)

(RuleUsing DETERQ is NP3)

(Rule for DETERQ

(DETERQ1 DETERQ = (DETQ # 1) ER Q ;
CONSTRUCTOR

(PROGN (COND
((@ DETQ # 1)
(OR (AGREE NBR DETQ # 1 Q)
(F.REJECT (QUOTE F.NBR))))))
(@FROM Q NBR TYPE)
(@SET THANCOMP T)))
)

(Category DETQ)

(RulesUsing DETQ are ADJP2 ADVP2 DETERQ1 NP1 and NP2)

(Rules for DETQ

(DETQ1 DETQ = (A) Q (QPP) ;
CONSTRUCTOR

(PROGN (@FROM Q TYPE NBR)
(OR (NOT (@ QPP))
(AGREE NBR Q QPP)
(F.REJECT (QUOTE F.NBRQ))))
)

(DETQ2 DETQ = NUMBER ;
CONSTRUCTOR

(@FROM NUMBER NBR))

(DETQ3 DETQ = ("THE") EST Q ;
CONSTRUCTOR

(PROGN (@FROM Q NBR TYPE)
(AND (@ THE)
(@SET DEF T)))
)

(Category DO)

(RulesUsing DO are DOP1 and SDQ1)

(Category DOP)

(RuleUsing DOP is AUXD1)

(Rule for DOP

(DOP1 DOP = (ADV) ({ "-S" / PAST }) (DO) ;
CONSTRUCTOR

```

(PROGN (COND
  ((@ PAST)
    (@SET TENSE (QUOTE PAST)))
  ((@ -S)
    (@SET TENSE (QUOTE PRES))
    (@SET NBR (QUOTE SG))
    (@SET SUFFIXONLY T))
  (T (@SET TENSE (QUOTE PRES))
    (@SET NBR (QUOTE PL))
    (@SET INFINITIVE T))))
)

```

(Category ENOUGH)

(RulesUsing ENOUGH are ADJP3 and ADVP1)

(Category ER)

(RulesUsing ER are ADJP2 ADVNBR ADVP2 ASNP2 DETERQ1 NOMHD3 WHADJP1 and WHDET1)

(Category EST)

(RulesUsing EST are ADJP6X ADVP6X DETQ3 and NPEST)

(Category FOR)

(RulesUsing FOR are ADJCOMP INFREL and VP4)

(Category FROM)

(RuleUsing FROM is WHPP1)

(Category HAVE)

(RuleUsing HAVE is HAVEP1)

(Category HAVEP)

(RulesUsing HAVEP are AUXB1 AUXD1 GERUND INF1 SBQ2 SBQ3 SDQ2 SDQ3 STHEREQ2 and STHEREQ3)

(Rule for HAVEP

(HAVEP1 HAVEP = (ADV) ({"-S" / PAST}) HAVE ;
CONSTRUCTOR

```

(PROGN (COND
  ((@ PAST)
    (@SET TENSE (QUOTE PAST)))
  ((@ -S)
    (@SET TENSE (QUOTE PRES))
    (@SET NBR (QUOTE SG)))
  (T (@SET TENSE (QUOTE PRES))
    (@SET NBR (QUOTE PL))
    (@SET INFINITIVE T))))
)

```

(Category HOW)

(RulesUsing HOW are WHADJP1 WHDET1 and WHPP2)

(Category INFINITIVE)

(RulesUsing INFINITIVE are ADJCOMP INFREL VP4 and VP5)

(Rule for INFINITIVE

(INF1 INFINITIVE = (ADV TO (HAVEP PPL) (BEING) {VP / BEP
{VPASS / PRED}});
CONSTRUCTOR

(PROGN (AND (@ BEP
(OR (@ INFINITIVE BEP
(F REJECT (QUOTE F.INFBE))))
(AND (@ HAVEP
(OR (@ INFINITIVE HAVEP
(F REJECT (QUOTE F.INFHAVE))))))
)
)

(Category INFINITREL)

(RuleUsing INFINITREL is NCOMP1)

(Rule for INFINITREL

(INFREL INFINITREL = ({FOR NP / P RELPRO}) INFINITIVE;
CONSTRUCTOR

(PROGN)
)
)

(Category ING)

(RulesUsing ING are AUXD1 BEING1 GERUND MOD1 NOMHD2 SDQ2 SDQ3 SDQ4 SX4
VP2 and VPCONJ)

(Rule for ING

(ING1 ING = (ADV "ING");
)

(Category LIST)

(Category LITATOM)

(Category MOD)

(RuleUsing MOD is NCOMP1)

(Rule for MOD

(MOD1 MOD = ({ING / PPL} VP / {OF / P} NP);
CONSTRUCTOR

(PROGN (AND (@ NP)
(@FROM NP TIME))

```

(AND (@ BAREV VP)
  (F.REJECT (QUOTE F.BAREV))))
)
)

(Category MODAL)

(RuleUsing MODAL is MODAL)

(Category MODALP)

(RulesUsing MODALP are AUXB1 AUXD1 SBQ2 SDQ2 and STHEREQ2)

(Rule for MODALP

(MODAL MODALP = (ADV) MODAL ;
  CONSTRUCTOR

(PROGN (@FROM MODAL PDGM SEMNODE SEMSIT)))
)

(Category N)

(RulesUsing N are NOUN1 and NOUN2)

(Category NCOMP)

(RulesUsing NCOMP are NP1 NP2 NP3 NPEST and WHNP1)

(Rule for NCOMP

(NCOMP1 NCOMP = (MOD # 1 (MOD # 2)) (C = {MOD # 3 / INFINITREL / SREL
/ ADJP / THANCOMP}));
  CONSTRUCTOR

(PROGN (COND
  ((OR (AND (@ TIME MOD # 1)
    (@ TIME MOD # 2))
    (AND (@ TIME MOD # 2)
    (@ TIME MOD # 3)))
    (F.REJECT (QUOTE F.NCOMP1-MOD))))
  (@FROM SREL NBR)
  (COND
    ((AND (@ MOD # 2)
    (@ C))
    (@FACTOR (QUOTE F.MOD3)
    UNLIKELY))
    (T (@FACTOR (QUOTE F.MOD2)
    FAIR)))
  (AND (@ MOD # 3)
    (OR (@ MOD # 2)
    (F.REJECT (QUOTE F.MOD))))
  (AND (@ ADJP)
    (@ NOCOMP ADJP)
    (F.REJECT (QUOTE F.NOCOMP)))
  (AND (@ SREL)
    (@ MOD # 1)
    (@FACTOR (QUOTE F.NCOMP1)
    UNLIKELY)))
  INTEGRATOR

```

(INT.NCOMP1))
)

(Category NOMHD
CONSTRUCTOR (PROGN))

(RulesUsing NOMHD are ASNP NOMHD1 NOMHD2 NOMHD3 NP1 NP3 NPEST and WHNP1)

(Rules for NOMHD

(NOMHD1 NOMHD = NOUN ({(",") "AND" / ","} NOMHD) ;
CONSTRUCTOR

(PROGN (COND
((@ NOMHD)
(@SET TIME (OR (@ TIME NOUN)
(@ TIME NOMHD)))
(@SET NBR (QUOTE PL))
(COND
(MASS? NOUN)
(@SET TYPE (QUOTE COMMON)))
(T (@FROM NOUN TYPE))))
(T (@FROM NOUN TYPE NBR TIME)))
(@FROM NOUN NUMVAL))
)

(NOMHD2 NOMHD = {ADJ / {ING / PPL} V (NUMBER)} NOMHD ;
CONSTRUCTOR

(PROGN (COND
((@ PPL)
(OR (@ DIROBJ V)
(F.REJECT (QUOTE F.DIROBJ))))
(@FROM NOMHD NBR TYPE TIME))
)

(NOMHD3 NOMHD = (QPP) ER ("MUCH") ADJ NOMHD ;
CONSTRUCTOR

(PROGN (COND
((@ QPP)
(OR (EQ (@ NBR QPP)
(QUOTE SG))
(F.REJECT (QUOTE F.NBRQ))))
(@FROM NOMHD NBR TYPE NUMVAL)
(@SET THANCOMP T)))
)

(Category NOT)

(RulesUsing NOT are SBQ1 SBQ2 SBQ3 SDQ1 SDQ2 SDQ3 SDQ4 SIMP2 STHEREQ1
STHEREQ2 STHEREQ3 and TAGAUX)

(Category NOUN
CONSTRUCTOR (PROGN))

(RulesUsing NOUN are NOMHD1 and NOUN2)

(Rules for NOUN

(NOUN1 NOUN = ("S") N ;
CONSTRUCTOR

(PROGN (@FROM N TYPE NUMVAL TIME UNIT)
(COND
((@ -S)
(@SET NBR (QUOTE PL))
(* FOR MASS TYPE NOUNS,
CONVERT TO "KINDS OF"
INTERPRETATION.)
(AND (MASS? N)
(@SET TYPE (QUOTE COMMON))))
(T (@SET NBR (QUOTE SG))))))
)

(NOUN2 NOUN = N NOUN ;
CONSTRUCTOR

(@FROM NOUN TIME NBR UNIT))
)

(Category NP
CONSTRUCTOR (AND (@ NOMHD)
(@FROM NOMHD TIME TYPE)))

(RulesUsing NP are ADJCOMP ASCOMP ASNP ASNP2 DDET2 INFREL MOD1 PP1 PPX
PRED1 RESPONSE2 SB1 SBQ1 SBQ2 SBQ3 SD1 SDQ1
SDQ2 SDQ3 SDQ4 STHERE1 STHEREQ1 STHEREQ2
STHEREQ3 THANCOMP VP1 VP2 VP3 VP4 and VP5)

(Rules for NP

(ASNP NP = AS QPP ({OF NP / NOMHD}) (ASCOMP) ;
CONSTRUCTOR

(PROGN (@FROM QPP NBR TYPE)
(OR (NOT (@ NOMHD))
(AGREE NBR QPP NOMHD)
(F.REJECT (QUOTE F.NBR))))))

(ASNP2 NP = AS ((QPP) ER ("MUCH")) ADJ (OF) NP (THANCOMP) (ASCOMP) ;
CONSTRUCTOR

(PROGN (COND
((@ QPP)
(OR (EQ (@ NBR QPP)
(QUOTE SG))
(F.REJECT (QUOTE F.NBRQ))))))
(COND
((@ THANCOMP)
(OR (@ ER)
(F.REJECT (QUOTE F.THANCOMP))))))
(@FROM NP NBR TIME)))

(GERUND NP = (DDET) ING (HAVEP PPL) {VP / BE {PRED / VPASS}} ;
CONSTRUCTOR

(PROGN (COND
((AND (@ DDET)

```

(OR (@ NP VP)
  (@ NP # 1 VP)))
(@FACTOR (QUOTE F.GERUND)
  FAIR))
((@ GEN DDET)
  (@FACTOR (QUOTE F.GERUND)
    PROBABLE)))
(@SET NBR 'SG)
(AND (@ CONJ VP)
  (F.REJECT (QUOTE F.GERUND))))

```

```

(NP1 NP = (D = {A / DDET / DETQ}) NOMHD (NCOMP) ;
  CONSTRUCTOR

```

```

(PROGN (COND
  ((@ D)
    (AND (EQ (QUOTE PROPERNAME)
      (@ TYPE NOMHD))
      (F.REJECT (QUOTE F.NP1-PROPERNAME))))
    (COND
      ((MASS? D)
        (OR (MASS? NOMHD)
          (F.REJECT (QUOTE F.MASS))))))
    (COND
      ((@ NCOMP)
        (@SET NBR (@INTERSECT NBR D NOMHD NCOMP)))
        (T (@SET NBR (@INTERSECT NBR D NOMHD))))
        (@FROM D DEF)
        (@FACTOR (QUOTE F.NP1-DET)
          LIKELY))
      ((AND (SG? NOMHD)
        (EQ (@ TYPE NOMHD)
          (QUOTE COMMON)))
        (@FACTOR (QUOTE F.NODET)
          UNLIKELY))
      ((@ NCOMP)
        (@SET NBR (@INTERSECT NBR NCOMP NOMHD)))
        (T (@FROM NOMHD NBR)))
      (AND (@ THANCOMP NCOMP)
        (OR (@ THANCOMP NOMHD)
          (F.REJECT (QUOTE F.THANC))))
      (AND (EQ (QUOTE PROPERNAME)
        (@ TYPE NOMHD))
        (NOT (@ D))
        (SG? NOMHD)
        (@ NCOMP)
        (F.REJECT (QUOTE PROPERNOUN-NP1))))
  )

```

```

(NP2 NP = D = {A / DDET / DETQ} (NCOMP) ;
  CONSTRUCTOR

```

```

(PROGN (AND (@ INCOMPLETE D)
  (F.REJECT (QUOTE F.INCOMPLETE)))
  (AND (@ THANCOMP NCOMP)
    (OR (AND (@ THANCOMP D)
      (AGREE TYPE THANCOMP D))
      (F.REJECT (QUOTE F.THANCOMP))))
  (COND
    ((@ NCOMP)

```

```

(AND (@ SREL NCOMP)
  (NOT (@ RELPRO SREL))
  (F.REJECT (QUOTE F.NCOMP)))
(@INTERSECT NBR NCOMP D))
(T (@FROM D NBR)))
(@FROM D DEF)
(OR (@ NCOMP)
  (@FACTOR (QUOTE F.NP2-NCOMP)
    UNLIKELY)))
)

```

```

(NP3 NP = DETERQ (NOMHD) (NCOMP) ;
  CONSTRUCTOR

```

```

(PROGN (COND
  ((@ NOMHD)
    (AND (EQ (@ TYPE NOMHD)
      (QUOTE PROPERNAME))
      (F.REJECT (QUOTE F.NP3-PROPERNAME)))
    (OR (AGREE NBR DETERQ NOMHD)
      (F.REJECT (QUOTE F.NBR)))
    (@FROM NOMHD TYPE))
  ((@ NCOMP)
    (AND (@ THANCOMP NCOMP)
      (F.REJECT (QUOTE F.NP3-THAN)))
    (OR (AGREE NBR DETERQ NCOMP)
      (F.REJECT (QUOTE F.NBR)))
    (T (@FROM DETERQ TYPE)))
  (COND
    ((@ THANCOMP NCOMP)
      (AND (MASS? DETERQ)
        (OR (MASS? THANCOMP)
          (F.REJECT (QUOTE F.MASS))))
    (@FROM DETERQ NBR)))

```

```

(NP5 NP = PRO ;
  CONSTRUCTOR

```

```

(PROGN (@FROM PRO NOMCASE NBR)
  (@SET DEF T))
)

```

```

(NPEST NP = (DDET) EST ("MUCH") ADJ (NOMHD) (NCOMP) ;
  CONSTRUCTOR

```

```

(PROGN (AND (@ THANCOMP NCOMP)
  (OR (@ THANCOMP NOMHD)
    (F.REJECT (QUOTE F.NCOMP)))
  (COND
    ((AND (@ DDET)
      (@ NOMHD))
      (OR (AGREE NBR DDET NOMHD)
        (F.REJECT (QUOTE F.NBR)))
      (@SET NBR (@INTERSECT NBR DDET NOMHD)))
    (T (@FROM DDET NBR))))))

```

```

(NPX NP = NUMBER {"A.M" / "P.M"} ;
  CONSTRUCTOR

```

```

(PROGN (@SET TIME T))

```

)

(Category NUMBER)

(RulesUsing NUMBER are ADVNBR DDET1 DETQ2 NOMHD2 NPX and WHNP1)

(Rule for NUMBER

(ADV NBR NUMBER = {ER "MANY" "THAN" / ADV} NUMBER ;
CONSTRUCTOR

(PROGN (@FROM NUMBER NUMVAL)))

)

(Category NUMERAL)

(Category OF)

(RulesUsing OF are ADJCOMP ASNP ASNP2 and MOD1)

(Category P)

(RulesUsing P are INFREL MOD1 MODP PP1 PP5 SREL1 VP1 and WHPP1)

(Rule for P

(MODP P = ADV P ;
CONSTRUCTOR

(@FROM P INGCOMP SOBJ PREPCOMP PDGM))

)

(Category PAST)

(RulesUsing PAST are DOP1 HAVEP1 and SDQ1)

(Rule for PAST

(PAST PAST = "ED" ;)

)

(Category PP

CONSTRUCTOR (AND (@ NP)
(@FROM NP TIME)))(RulesUsing PP are PPZ PRED1 PRED2 STHERE1 STHEREQ1 STHEREQ2 STHEREQ3
SWH3 SX3 and VP6)

(Rules for PP

(PP1 PP = (P) (NP) ;
CONSTRUCTOR

(PROGN (AND (@ NP)

(@ INTRANSITIVE P)

(F.REJECT 'F.PP1-INTRANS))

(OR (@ P)

(@ TIME NP)

(F.REJECT (QUOTE F.PP1)))

(AND (@ P)

```

    (@ ING NP)
    (@ INGCOMP P)
    (@FACTOR 'F.PP1-GERUND PROBABLE))
  (OR (@ NP)
    (@SET BAREPREP T))
  (AND (@ P)
    (@ NP)
    (@FACTOR 'F.PP1-COMplete LIKELY))
  (COND
    ((@ NP)
      (OR (NOT (@ NOMCASE NP))
        (F.REJECT (QUOTE F.CASEPP1)))
      (@FROM NP TIME))))
)

```

```

(PP5  PP = (P) ADV ;
  CONSTRUCTOR

```

```

(PROGN (OR (@ P)
  (@SET BAREPREP T))
  (OR ('NOT (@ P))
    (@ DEIXIS ADV)
    (F.REJECT 'F.PP5.DEIXIS))
  (@FACTOR (QUOTE F.PP5)
    UNLIKELY))
)

```

```

(PPSUBCONJ
  PP = SUBCONJ SDEC ;
  CONSTRUCTOR

```

```

(PROGN))

```

```

(PPX  PP = NP "AGO" ;
  CONSTRUCTOR

```

```

(PROGN (OR (AND (@ TIME NP)
  (EQ (@ TYPE NP)
    (QUOTE UNIT)))
  (F.REJECT (QUOTE F.PPX)))
  (@SET TIME T))
)

```

```

(PPZ  PP = PP #1 PP #2 ;
  CONSTRUCTOR

```

```

(PROGN (OR (AND (@ TIME PP #1)
  (@ TIME PP #2))
  (F.REJECT (QUOTE F.PPZ)))
  (AND (@ PP PP #1)
    (F.REJECT 'F.PPZ-L-RECURS))
  (@FACTOR 'F.PPZ.TIME LIKELY)))
)

```

```

(Category PPL)

```

```

(RulesUsing PPL are AUXB1 AUXD1 BEEN GERUND INF1 MOD1 NOMHD2 SBQ2 SDQ2
  SDQ3 STHEREQ2 SX4 VP2 VPASS1 and VPCONJ)

```

```

(Rule for PPL

```

(PPL1 PPL = (ADV) {"EN" / "ED"} ;)
)

(Category PRED)

(RulesUsing PRED are GERUND INF1 PRED2 SB1 SBQ1 SBQ2 SBQ3 SIMP2 and
 SREL2)

(Rules for PRED

(PRED1 PRED = {ADJP / (ADV) NP / PP} ;
 CONSTRUCTOR

(PROGN (COND
 ((@ ING NP)
 (OR (@ DDET NP)
 (F.REJECT (QUOTE F.PRED1-GERUND))))))
 (AND (@ PP)
 (OR (@ VP PP)
 (@ PRED PP)
 (@ SDEC PP)
 (@ VPASS PP))
 (F.REJECT 'F.PRED1))
 (AND (@ ADJP)
 (@ CONJ ADJ ADJP)
 (F.REJECT 'F.PREDADJ))))

(PRED2 PRED = PRED ("," PP ;
 CONSTRUCTOR

(PROGN (AND (@ TIME PP PRED)
 (@ TIME PP)
 (F.REJECT 'F.PRED2-PP)))
)

(Category PRO)

(RulesUsing PRO are NP5 and TAGAUX)

(Category Q
 CONSTRUCTOR (@SET NUMVAL (@ DIAMOND.SPELLING)))

(RulesUsing Q are DETERQ1 DETQ1 DETQ3 QPP1 and WHDET1)

(Category QPP)

(RulesUsing QPP are ASNP ASNP2 DDET1 DETQ1 NOMHD3 QPP1 WHADJP1 and
 WHDET1)

(Rule for QPP

(QPP1 QPP = (ADV) Q (QPP) ;
 CONSTRUCTOR

(PROGN (@FROM QPP NBR TYPE)))
)

(Category RELPRO)

(RulesUsing RELPRO are INFREL SREL1 and SREL2)

(Category RESPONSE)

(Rules for RESPONSE

(RESPONSE1

RESPONSE = {TAGWORD / ADVP / ADJP} {"," / "!" / "?"} ;)

(RESPONSE2

RESPONSE = "HOW" "ABOUT" NP "?" ;)

)

(Category S

CONSTRUCTOR (PROGN)

INTEGRATOR (GOAL-CONTROL ROOT))

(RulesUsing S are S0 SX1 SX2 SX3 and SX4)

(Rules for S

(S0 S = (TAGWORD ",") {S C = {"," / "!" / "?" / TAG} / SDEC
/ SIMP / SQ / SWHQ} ;

CONSTRUCTOR

(COND

((@ SDEC)

(@SET SENTYPE (QUOTE SDEC)))

((@ SIMP)

(@SET SENTYPE (QUOTE SIMP)))

((@ SQ)

(@SET SENTYPE (QUOTE SQ)))

((@ SWHQ)

(@SET SENTYPE (QUOTE SWHQ)))

((@ S)

(@FROM S SENTYPE))

((OR (EQ (QUOTE SWHQ)

(@ SENTYPE S # 1))

(EQ (QUOTE SWHQ)

(@ SENTYPE S # 2)))

(@SET SENTYPE (QUOTE SWHQ)))

((OR (EQ (QUOTE SQ)

(@ SENTYPE S # 1))

(EQ (QUOTE SQ)

(@ SENTYPE S # 2)))

(@SET SENTYPE (QUOTE SQ)))

((OR (EQ (QUOTE SIMP)

(@ SENTYPE S # 1))

(EQ (QUOTE SIMP)

(@ SENTYPE S # 2)))

(@SET SENTYPE (QUOTE SIMP)))

(T (@SET SENTYPE (QUOTE SDEC))))

)

(SX1 S = CONJ (",") S ;

CONSTRUCTOR

(PROGN (AND (@ SIMP S)

(F.REJECT 'F.S '1-SIMP))

(@SET CONJUNCT (@ CONJ))

(@FROM S WHQ IND IMP YESNO)

```
(AND (@ C S)
  (F.REJECT (QUOTE F.SX1C))))
```

```
(SX2  S = S # 1 (D = {" / ";}) S # 2 ;
  CONSTRUCTOR
```

```
(PROGN (OR (@ D)
  (@ CONJ S # 2)
  (F.REJECT (QUOTE F.CSX2)))
  (AND (@ C S # 2)
    (F.REJECT 'F.SX2C))
  (COND
    ((@ WHQ S # 1)
      (OR (@ WHQ S # 2)
        (@ .)
        (@ .)
        (F.REJECT (QUOTE F.NOPARSX2))))
    (AND (@ SIMP S # 2)
      (F.REJECT (QUOTE F.SIMP-SX2))))
```

```
(SX3  S = PP (" ") S ;
  CONSTRUCTOR
```

```
(PROGN (AND (@ TIME PP S)
  (@ TIME PP)
  (F.REJECT 'SX3-PP))
  (AND (@ C S)
    (F.REJECT (QUOTE F.SX3C)))
  (AND (@ PP S)
    (F.REJECT (QUOTE F.TWOPP)))
  (AND (@ BAREPREP PP)
    (NOT (@ ADV PP))
    (F.REJECT (QUOTE F.BAREPP)))
  (AND (EQ (@ DIAMOND.SPELLING PP)
    (QUOTE THERE))
    (@ SQ S)
    (F.REJECT (QUOTE F.THERE))))
```

```
(SX4  S = {PPL / ING} VP " " S ;
  CONSTRUCTOR
```

```
(AND (@ C S)
  (F.REJECT 'F.SX4C)))
)
```

```
(Category SDEC)
```

```
(RulesUsing SDEC are ASCOMP PPSUBCONJ S0 SREL1 THANCOMP and VP3)
```

```
(Rules for SDEC
```

```
(SB1  SDEC = NP AUXB ({PRED / VPASS}) ;
  CONSTRUCTOR
```

```
(PROGN (OR (AGREE NBR NP AUXB)
  (F.REJECT (QUOTE F.NBR)))
  (@FROM AUXB TENSE PROGRESSIVE PERFECTIVE MODALITY))
)
```

```
(SD1  SDEC = NP (AUXD) VP ;
```


CONSTRUCTOR

```
(PROGN (COND
  ((@ AUXD)
    (@FROM AUXD TENSE PROGRESSIVE PERFECTIVE MODALITY)
    (OR (AGREE NBR NP AUXD)
      (F.REJECT (QUOTE F.NBRSD1))))))
  (T (OR (NEQ (@ NBR NP)
    (QUOTE SG))
    (EQ (@ NBR VP)
      (QUOTE SG))
    (F.REJECT (QUOTE F.NBR))))))
)
```

```
(STHERE1 SDEC = "THERE" AUXB {NP (",") (PP # 1 (PP # 2)) / VPASS} ;
  CONSTRUCTOR
```

```
(PROGN (AND (@ TIME PP # 1)
  (@ TIME PP # 2)
  (F.REJECT 'F.STHERE1-PP))
  (OR (NOT (@ VPASS))
    (@ INFOBJ VP VPASS)
    (F.REJECT 'F.STH1-VPASS))
  (OR (AGREE NBR AUXB NP)
    (F.REJECT (QUOTE F.NBRSTHERE1))))
  (AND (@ DIRECTION PP # 2)
    (@ DIRECTION PP # 1)
    (F.REJECT (QUOTE F.PPDIR)))
  (@FROM AUXB TENSE PROGRESSIVE PERFECTIVE MODALITY))
)
```

```
(Category SIMP)
```

```
(RuleUsing SIMP is S0)
```

```
(Rules for SIMP
```

```
(SIMP1 SIMP = ("DO") VP ;)
```

```
(SIMP2 SIMP = ("DO" (NOT)) "BE" {PRED / VPASS} ;
  CONSTRUCTOR
```

```
(PROGN))
```

```
)
```

```
(Category SQ)
```

```
(RulesUsing SQ are S0 and SWH1)
```

```
(Rules for SQ
```

```
(SBQ1 SQ = BEP (NOT) NP (BEING) ({PRED / VPASS}) ;
  CONSTRUCTOR
```

```
(PROGN (OR (AGREE NBR NP BEP)
  (F.REJECT (QUOTE F.NBRSBQ1)))
  (@FROM BEP TENSE)
  (AND (@ BEING)
    (@SET PROGRESSIVE T)))
```

)

(SBQ2 SQ = MODALP (NOT) NP (HAVEP PPL) BEP (BEING) ({PRED / VPASS});

CONSTRUCTOR

(PROGN (OR (@ INFINITIVE BEP)
 (F.REJECT (QUOTE BEP)))
 (OR (NOT (@ HAVEP))
 (@ INFINITIVE HAVEP)
 (F.REJECT (QUOTE F.HAVESBQ2)))
 (AND (@ BEING)
 (@SET PROGRESSIVE T))
 (AND (@ HAVEP)
 (@SET PERFECTIVE T)
 (@SET MODALITY T)))

)

(SBQ3 SQ = HAVEP (NOT) NP BEEN (BEING) ({PRED / VPASS});

CONSTRUCTOR

(PROGN (OR (AGREE NBR NP HAVEP)
 (F.REJECT (QUOTE F.NBR SBQ3)))
 (AND (@ BEING)
 (@SET PROGRESSIVE T))
 (@FROM HAVEP TENSE)
 (@SET PERFECTIVE T))

)

(SDQ1 SQ = ({"-S" / PAST}) DO (NOT) NP VP ;

CONSTRUCTOR

(PROGN (OR (NOT (@ -S))
 (NEQ (@ NBR NP)
 (QUOTE PL))
 (F.REJECT (QUOTE F.NBR SDQ1)))
 (@SET TENSE (COND
 ((@ PAST)
 (QUOTE PAST))
 (T (QUOTE PRES)))))

)

(SDQ2 SQ = MODALP (NOT) NP (HAVEP PPL) (BEP ING) VP ;

CONSTRUCTOR

(PROGN (AND (@ TENSE BEP)
 (F.REJECT (QUOTE F.BINF)))
 (@SET MODALITY T)
 (AND (@ HAVEP)
 (OR (@ INFINITIVE HAVEP)
 (F.REJECT (QUOTE H.INF)))
 (@SET PERFECTIVE T))
 (AND (@ ING)
 (@SET PROGRESSIVE T)))

)

(SDQ3 SQ = HAVEP (NOT) NP PPL (BEP ING) VP ;

CONSTRUCTOR

(PROGN (OR (AGREE NBR HAVEP NP)

```

(F.REJECT (QUOTE F.NBRSDQ3)))
(AND (@ TENSE BEP)
(F.REJECT (QUOTE F.BINF)))
(@FROM HAVEP TENSE)
(@SET PERFECTIVE T)
(AND (@ ING)
(@SET PROGRESSIVE T)))
)

```

```

(SDQ4 SQ = BEP (NOT) NP ING VP ;
CONSTRUCTOR

```

```

(PROGN (OR (AGREE NBR BEP NP)
(F.REJECT (QUOTE F.NBRSDQ3)))
(@SET YESNO T)
(@SET PROGRESSIVE T)
(@FROM BEP TENSE))
)

```

```

(STHEREQ1
SQ = BEP (NOT) "THERE" (NP) ("") (PP # 1 (PP # 2)) ;
CONSTRUCTOR

```

```

(PROGN (AND (@ TIME PP # 1)
(@ TIME PP # 2)
(F.REJECT 'F.STHEREQ1-PP))
(OR (NOT (@ NP))
(AGREE NBR NP BEP)
(F.REJECT (QUOTE F.NBRTHEREQ1)))
(@FROM BEP TENSE))
)

```

```

(STHEREQ2
SQ = MODALP (NOT) "THERE" (HAVEP PPL) BE (NP) ("")

```

```

(PP # 1 (PP # 2)) ;
CONSTRUCTOR

```

```

(PROGN (AND (@ TIME PP # 1)
(@ TIME PP # 2)
(F.REJECT 'F.STHEREQ2-PP))
(AND (@ TENSE BE)
(F.REJECT (QUOTE F.BINF)))
(AND (@ HAVEP)
(@SET PERFECTIVE T))
(@SET MODALITY T))
)

```

```

(STHEREQ3
SQ = HAVEP (NOT) "THERE" BEEN (NP) ("") (PP # 1 (PP # 2)) ;
CONSTRUCTOR

```

```

(PROGN (AND (@ TIME PP # 1)
(@ TIME PP # 2)
(F.REJECT 'F.STHEREQ3-PP)))
)
)

```

```

(Category SREL)

```

(RuleUsing SREL is NCOMP1)

(Rules for SREL

(SREL1 SREL = ((P) RELPRO) SDEC ;
CONSTRUCTOR

(PROGN (AND (@ P)
 (@ THAT RELPRO)
 (F.REJECT (QUOTE F.THAT)))
(OR (@ VP SDEC)
 (@ VPASS SDEC)
 (@ C SDEC)
 (F.REJECT (QUOTE F.ELLIPSE)))))

(SREL2 SREL = RELPRO {AUXB {PRED / VPASS} / (AUXD) VP} ;
CONSTRUCTOR

(PROGN (COND
 ((@ AUXB)
 (@FROM AUXB NBR TENSE PROGRESSIVE PERFECTIVE MODALITY))
 ((@ AUXD)
 (@FROM AUXD NBR TENSE PROGRESSIVE PERFECTIVE MODALITY))
 (T (@SET NBR (QUOTE PL)))))
)

(Category STRING)

(Category SUBCONJ)

(RulesUsing SUBCONJ are PPSUBCONJ and SUBCONJ1)

(Rule for SUBCONJ

(SUBCONJ1
 SUBCONJ = ADV SUBCONJ ;
)

(Category SWHQ)

(RuleUsing SWHQ is S0)

(Rules for SWHQ

(SWH1 SWHQ = {WHNP / WHPP / WHADJP} SQ ;
CONSTRUCTOR

(PROGN (COND
 ((@ WHNP)
 (AND (OR (@ NP # 2 SQ)
 (@ ADJP SQ))
 (F.REJECT (QUOTE F.2PRED)))))
 (AND (@ WHADJP)
 (@ ADJP SQ)
 (F.REJECT (QUOTE F.ADJPRE)))
 (@FROM SQ TENSE PROGRESSIVE PERFECTIVE MODALITY))
)

(SWH2 SWHQ = WHNP (AUXD) VP ;
CONSTRUCTOR

```

(PROGN (COND
  ((@ AUXD)
    (OR (AGREE NBR AUXD WHNP)
      (F.REJECT (QUOTE F.NBR)))
    (@FROM AUXD TENSE PROGRESSIVE PERFECTIVE MODALITY))
  (T (OR (NEQ (@ NBR WHNP)
    (QUOTE SG))
    (F.REJECT (QUOTE F.NBR2))))))
)

(SWH3 SWHQ = WHNP AUXB ({C = {ADJP / PP # 1} (PP # 2 (PP # 3)) / VPASS}} ;
  CONSTRUCTOR

(PROGN (AND (OR (AND (@ TIME PP # 1)
  (@ TIME PP # 2))
  (AND (@ TIME PP # 2)
    (@ TIME PP # 3)))
  (F.REJECT 'F.SWH3-PP))
  (OR (EQ (@ TYPE WHNP)
    (QUOTE UNIT))
  (AGREE NBR WHNP AUXB)
  (F.REJECT (QUOTE F.NBR)))
  (@FROM AUXB TENSE PROGRESSIVE PERFECTIVE MODALITY))
)
)

(Category TAG)

(RuleUsing TAG is S0)

(Rules for TAG

(TAGAUX TAG = "," {AUXB / AUXD} (NOT) {PRO / "THERE"} "?" ;)

(TAGWD TAG = "," TAGWORD "?" ;)
)

(Category TAGWORD)

(RulesUsing TAGWORD are RESPONSE1 S0 and TAGWD)

(Category THANCOMP)

(RulesUsing THANCOMP are ADJP2 ADVP2 ASNP2 NCOMP1 and WHADJP1)

(Rule for THANCOMP

(THANCOMP
  THANCOMP = "THAN" {NP / SDEC} ;
  CONSTRUCTOR

(PROGN (@FROM NP TYPE)
  (AND (@ NP)
    (@ NUMVAL NP)
    (F.REJECT 'F.THANCOMP)))
)

(Category TO)

```

(RulesUsing TO are INF1 and WHPP1)

(Category V)

(RulesUsing V are ADVV NOMHD2 VP1 VP2 VP3 VP4 and VP5)

(Rule for V

(ADV V = ADV V ;
CONSTRUCTOR

(PROGN (COND
((EQ (@ DIAMOND.SPELLING ADV)
(QUOTE NOT))
(@SET NEG T)))
(@FROM V DIROBJ INDIROBJ INFOBJ PREDOBJ SOBJ ABSTSUBJ PARTICLE
INSEPARABLE PRECOMP NO.PASSIVE PDGM SEMNODE LIKE)))
)

(Category VP

CONSTRUCTOR (PROGN (COND
((OR (AND (@ NP # 1)
(@ NOMCASE NP # 1))
(AND (@ NP # 2)
(@ NOMCASE NP # 2))
(AND (@ NP # 3)
(@ NOMCASE NP # 3))
(AND (@ NP)
(@ NOMCASE NP))))
(F.REJECT (QUOTE F.VPNOM))))
(COND
((@ V)
(@FROM V INFOBJ NEG PREPCOMP))
((@ VP)
(@FROM VP NO.PASSIVE INFOBJ NEG PREPCOMP))
((@ VP # 1)
(@FROM VP # 1 INFOBJ NEG))))))

(RulesUsing VP are GERUND INF1 MOD1 SD1 SDQ1 SDQ2 SDQ3 SDQ4 SIMP1 SREL2
SWH2 SX4 VP2 VP6 VPASS1 and VPCONJ)

(Rules for VP

(VP1 VP = V (P # 1) (NP # 1 (P # 2) (NP # 2)) ;
CONSTRUCTOR

(PROGN (COND
((AND (@ P # 1)
(OR (@ P # 2)
(@ NP # 2)))
(F.REJECT (QUOTE F.VP1.P.P))))
(COND
((@ P # 1)
(OR (FMEMB (@ DIAMOND.SPELLING P # 1)
(@ PARTICLE V))
(FMEMB (@ DIAMOND.SPELLING P # 1)
(@ INSEPARABLE V))
(F.REJECT (QUOTE F.DIROBJVP1)))
(AND (@ NP # 1)
(@ PRO NP # 1)

```

(OR (FMEMB (@ DIAMOND SPELLING P # 1)
      (@ INSEPARABLE V))
  (F.REJECT (QUOTE F.PRO))))
(@FACTOR (QUOTE VP1-P1)
  PROBABLE))
((@ P # 2)
  (OR (FMEMB (@ DIAMOND SPELLING P # 2)
        (@ PARTICLE V))
    (F.REJECT (QUOTE F.VP1-P2)))
  (AND (@ NP # 2)
    (@ PRO NP # 2)
    (F.REJECT (QUOTE F.VP1-PRO2)))
  (@FACTOR (QUOTE VP1-P2)
    PROBABLE)))
(COND
  ((@ NP # 1)
    (OR (@ DIROBJ V)
      (F.REJECT (QUOTE F.DIROBJVP1)))
    (COND
      ((@ NP # 2)
        (OR (@ INDIROBJ V)
          (F.REJECT (QUOTE F.INDIROBJVP1))))))
  (AND (NOT (OR (@ P # 1)
    (@ NP # 1)))
    (@SET BAREV T))
  (COND
    ((OR (@ NP # 2)
      (AND (@ NP # 1)
        (NOT (@ INDIROBJ V))
        (NOT (@ DIROBJ V))
        (@ NO.PASSIVE V))
      (@SET NO.PASSIVE T))))
)

```

VP2 VP = V (NP) ({(ING / PPL)} VP / ADJP) ;
CONSTRUCTOR

```

(PROGN (@SET NO.PASSIVE (@ NP))
  (OR (@ DIROBJ V)
    (F.REJECT (QUOTE F.DIROBJ)))
  (OR (@ PREDOBJ V)
    (F.REJECT (QUOTE F.PREDOBJ)))
  (COND
    ((@ PPL)
      (OR (@ NP)
        (F.REJECT (QUOTE F.NOOBJ))))))
INTEGRATOR

```

(INT.VP))

VP3 VP = V (NP) ({WHPP / WHNP / "IF" / "THAT"}) SDEC ;
CONSTRUCTOR

```

(PROGN (OR (@ SOBJ V)
  (F.REJECT (QUOTE F.SOBJ)))
  (@FACTOR 'F.VP3 LIKELY)
  (@SET NO.PASSIVE (@ NP))
  (COND
    ((@ NP)
      (OR (@ DIROBJ V)

```

(F.REJECT (QUOTE F.DIROBJ))))))
)

(VP4 VP = V (NP # 1 (NP # 2)) ("IN" "ORDER") (FOR NP # 3) INFINITIVE ;
 CONSTRUCTOR

(PROGN (@SET NO.PASSIVE (@ NP # 1))
 (OR (@ INFOBJ V)
 (F.REJECT (QUOTE F.INFOBJ)))
 (COND
 ((@ NP # 1)
 (OR (@ DIROBJ V)
 (F.REJECT (QUOTE F.DIROBJ))))))
 (COND
 ((@ NP # 2)
 (OR (@ INDIROBJ V)
 (F.REJECT (QUOTE F.INDIROBJ))))))
 (COND
 ((@ NP # 3)
 (OR (@ ABSTSUBJ V)
 (F.REJECT (QUOTE F.ABSTSUBJ))))))
)

(VP5 VP = V (NP) {WHPP / WHNP} INFINITIVE ;
 CONSTRUCTOR

(PROGN (@SET NO.PASSIVE (@ NP))
 (OR (@ INFOBJ V)
 (F.REJECT (QUOTE F.INFOBJ)))
 (COND
 ((@ NP)
 (OR (@ DIROBJ V)
 (F.REJECT (QUOTE F.DIROBJ))))))
 INTEGRATOR

(INT.VP5))

(VP6 VP = VP ("," PP ;
 CONSTRUCTOR

(PROGN (AND (@ TIME PP VP)
 (@ TIME PP)
 (F.REJECT 'F.VP6-PP))
 (AND (OR (@ SDEC VP)
 (@ INFINITIVE VP))
 (@FACTOR (QUOTE VP6-PP)
 UNLIKELY))
 (AND (@ TIME PP)
 (@FACTOR 'FVP6 LIKELY))
 (AND (@ BAREV VP)
 (OR (FMEMB (@ DIAMOND.SPELLING P PP)
 (@ INSEPARABLE V VP))
 (FMEMB (@ DIAMOND.SPELLING P PP)
 (@ PARTICLE V VP)))
 (@FACTOR (QUOTE VP6-P)
 UNLIKELY))
 (AND (FMEMB (@ DIAMOND.SPELLING P PP)
 (@ PREPCOMP VP))
 (@FACTOR (QUOTE F.PREPCOMP)
 LIKELY))


```
(AND (@ .)
  (@ BAREPREP PP)
  (F.REJECT (QUOTE VP6.PP))))
)
```

```
(VPCONJ VP = (AUXD # 1) VP # 1 CONJ ({AUXD # 2 / PPL / ING}) VP # 2 ;
  CONSTRUCTOR
```

```
(PROGN (AND (OR (@ PPL)
  (@ ING)
  (EQ (QUOTE SG)
    (@ NBR AUXD # 2)))
  (NOT (@ AUXD # 1))
  (F.REJECT (QUOTE F.VPCONJ)))
  (@SET NO.PASSIVE T)
  (@SET CONJUNCT T)
  (@FROM AUXD # 1 NBR)
  (OR (@ AUXD # 1)
    (@ AUXD # 2)
    (@FACTOR (QUOTE F.VPCONJ)
      PROBABLE))))
)
```

```
(Category VPASS)
```

```
(RulesUsing VPASS are GERUND INF1 SB1 SBQ1 SBQ2 SBQ3 SIMP2 SREL2
  STHERE1 and SWH3)
```

```
(Rule for VPASS
```

```
(VPASS1 VPASS = PPL VP ;
  CONSTRUCTOR
```

```
(PROGN (AND (@ NO.PASSIVE VP)
  (F.REJECT (QUOTE F.VPASS)))
  (@FROM VP BAREV))
)
)
```

```
(Category WHADJP)
```

```
(RuleUsing WHADJP is SWH1)
```

```
(Rule for WHADJP
```

```
(WHADJP1 WHADJP = HOW (QPP ER ("MUCH")) ADJ (ADJCOMP) (THANCOMP) ;
  CONSTRUCTOR
```

```
(PROGN (COND
  ((@ THANCOMP)
    (OR (@ ER)
      (F.REJECT (QUOTE F.THANCOMP))))
  (COND
    ((@ QPP)
      (OR (EQ (@ NBR QPP)
        (QUOTE SG))
        (F.REJECT (QUOTE F.NBRQ))))))
)
)
```

(Category WHDET)

(RuleUsing WHDET is WHNP1)

(Rules for WHDET

(WHDET1 WHDET = HOW Q # 1 (QPP) (ER Q # 2) ;
CONSTRUCTOR

(PROGN (COND
((@ QPP)
(OR (AGREE NBR QPP Q # 1)
(F.REJECT (QUOTE F.NBRQ))))))
(OR (NOT (@ Q # 2))
(AGREE NBR Q # 1 Q # 2)
(F.REJECT (QUOTE F.NBRQ2)))
(@FROM Q # 1 NBR TYPE)))

(WHDET2 WHDET = WHNP "GEN" ;
CONSTRUCTOR

(PROGN))
)

(Category WHEN)

(Category WHERE)

(RuleUsing WHERE is WHPP1)

(Category WHILE)

(Category WHNP
CONSTRUCTOR (@SET CQUANT (QUOTE WH))
)

(RulesUsing WHNP are SWH1 SWH2 SWH3 VP3 VP5 WHDET2 and WHPP1)

(Rule for WHNP

(WHNP1 WHNP = WHDET (NUMBER) (NOMHD) (NCOMP) ;
CONSTRUCTOR

(PROGN (COND
((@ NUMBER)
(OR (NOT (@ NOMHD))
(AGREE NBR NOMHD NUMBER)
(F.REJECT (QUOTE F.NUMNBR)))
(@FROM NUMBER NBR)))
(COND
((@ NOMHD)
(@SET NBR (@INTERSECT NBR NOMHD WHDET))
(@FROM NOMHD TYPE))
(T (AND (@ SREL NCOMP)
(NOT (@ RELPRO SREL))
(F.REJECT (QUOTE F.NCOMP)))
(@FROM WHDET NBR TYPE))))))
)
)

(Category WHPP
)

(RulesUsing WHPP are SWH1 VP3 and VP5)

(Rules for WHPP

(WHPP1 WHPP = {P WHNP / (C = {TO / FROM}) WHERE} ;
CONSTRUCTOR

(PROGN (COND
((@ P
(@FROM P DIRECTION))
((@FROM C DIRECTION))))
)

(WHPP2 WHPP = HOW (ADV) ;
)

(Category LITERAL!)

(RulesUsing "!" are RESPONSE1 and S0)

(Words for LITERAL!
(!))

(Category LITERAL..)

(RulesUsing "." are NOMHD1 PRED2 S0 STHERE1 STHEREQ1 STHEREQ2 STHEREQ3
SX1 SX2 SX3 SX4 TAGAUX TAGWD and VP6)

(Words for LITERAL..
(.))

(Category LITERAL..S)

(RulesUsing "-S" are DOP1 HAVEP1 NOUN1 and SDQ1)

(Words for LITERAL..S
(-S))

(Category LITERAL..)

(RulesUsing "." are RESPONSE1 and S0)

(Words for LITERAL..
(%.))

(Category LITERAL.;)

(RuleUsing ";" is SX2)

(Words for LITERAL.;
(.))

(Category LITERAL.?)

(RulesUsing "?" are RESPONSE1 RESPONSE2 S0 TAGAUX and TAGWD)

(Words for LITERAL.?

(?)

(Category LITERAL.A.M)

(RuleUsing "A.M" is NPX)

(Words for LITERAL.A.M
(A.M))

(Category LITERAL.ABOUT)

(RuleUsing "ABOUT" is RESPONSE2)

(Words for LITERAL.ABOUT
(ABOUT))

(Category LITERAL.AGO)

(RuleUsing "AGO" is PPX)

(Words for LITERAL.AGO
(AGO))

(Category LITERAL.ALL)

(Words for LITERAL.ALL
(ALL))

(Category LITERAL.AM)

(Words for LITERAL.AM
(AM))

(Category LITERAL.AND)

(RuleUsing "AND" is NOMHD1)

(Words for LITERAL.AND
(AND))

(Category LITERAL.ANY)

(Words for LITERAL.ANY
(ANY))

(Category LITERAL.BE)

(RuleUsing "BE" is SIMP2)

(Words for LITERAL.BE
(BE))

(Category LITERAL.BETWEEN)

(Words for LITERAL.BETWEEN
(BETWEEN))

(Category LITERAL.CENT)

(Words for LITERAL.CENT

(CENT))

(Category LITERAL.DO)

(RulesUsing "DO" are SIMP1 and SIMP2)

(Words for LITERAL.DO
(DO))

(Category LITERAL.ED)

(RulesUsing "ED" are PAST and PPL1)

(Words for LITERAL.ED
(ED))

(Category LITERAL.EN)

(RuleUsing "EN" is PPL1)

(Words for LITERAL.EN
(EN))

(Category LITERAL.GEN)

(RulesUsing "GEN" are DDET2 and WHDET2)

(Words for LITERAL.GEN
(GEN))

(Category LITERAL.HENCE)

(Words for LITERAL.HENCE
(HENCE))

(Category LITERAL.HOW)

(RuleUsing "HOW" is RESPONSE2)

(Words for LITERAL.HOW
(HOW))

(Category LITERAL.IF)

(RuleUsing "IF" is VP3)

(Words for LITERAL.IF
(IF))

(Category LITERAL.IN)

(RuleUsing "IN" is VP4)

(Words for LITERAL.IN
(IN))

(Category LITERAL.ING)

(RuleUsing "ING" is ING1)

(Words for LITERAL.ING
(ING))

(Category LITERAL.LY)

(RuleUsing "LY" is ADV1)

(Words for LITERAL.LY
(LY))

(Category LITERAL.MANY)

(RuleUsing "MANY" is ADVNBR)

(Words for LITERAL.MANY
(MANY))

(Category LITERAL.MUCH)

(RulesUsing "MUCH" are ADJP2 ADJP6X ADVP2 ADVP6X ASNP2 NOMHD3 NPEST and
WHADJP1)

(Words for LITERAL.MUCH
(MUCH))

(Category LITERAL.NO)

(RuleUsing "NO" is ADJP2)

(Words for LITERAL.NO
(NO))

(Category LITERAL.NOW)

(Words for LITERAL.NOW
(NOW))

(Category LITERAL.ONLY)

(Words for LITERAL.ONLY
(ONLY))

(Category LITERAL.ORDER)

(RuleUsing "ORDER" is VP4)

(Words for LITERAL.ORDER
(ORDER))

(Category LITERAL.P.M)

(RuleUsing "P.M" is NPX)

(Words for LITERAL.P.M
(P.M))

(Category LITERAL.PER)

(Words for LITERAL.PER
(PER))

(Category LITERAL.PERCENT)

(Words for LITERAL.PERCENT
(PERCENT))

(Category LITERAL.PM)

(Words for LITERAL.PM
(PM))

(Category LITERAL.SO)

(Words for LITERAL.SO
(SO))

(Category LITERAL.SOME)

(Words for LITERAL.SOME
(SOME))

(Category LITERAL.THAN)

(RulesUsing "THAN" are ADVNBR and THANCOMP)

(Words for LITERAL.THAN
(THAN))

(Category LITERAL.THAT)

(RuleUsing "THAT" is VP3)

(Words for LITERAL.THAT
(THAT))

(Category LITERAL.THE)

(RuleUsing "THE" is DETQ3)

(Words for LITERAL.THE
(THE))

(Category LITERAL.THEN)

(Words for LITERAL.THEN
(THEN))

(Category LITERAL.THERE)

(RulesUsing "THERE" are STHERE1 STHEREQ1 STHEREQ2 STHEREQ3 and TAGAUX)

(Words for LITERAL.THERE
(THERE))

(Category LITERAL.TOO)

(Words for LITERAL.TOO
(TOO))

(Category LITERAL.VERY)

(Words for LITERAL.VERY
(VERY))

(Category LITERAL.WITHIN)

(Words for LITERAL.WITHIN
(WITHIN))
STOP

(SETROOTCATEGORIES (QUOTE (S RESPONSE)))

(SETQQ LASTSAVE (DCS)FROZEN.GRAMMAR;4 on 10-FEB-81% 10:21:10))
STOP

References

- [Robinson 82] Robinson, Jane J., "Diagram: A Grammar for Dialogues." *Communications of the ACM* 25, (1), January 1982, 27-47.
- [Walker 78] Walker, D. E. (ed.), *Understanding Spoken Language*, Elsevier North-Holland, NY, 1978.

FILMED

02 - 84